TWO-WAY DIGITAL PAGING SYSTEM USING SDRS

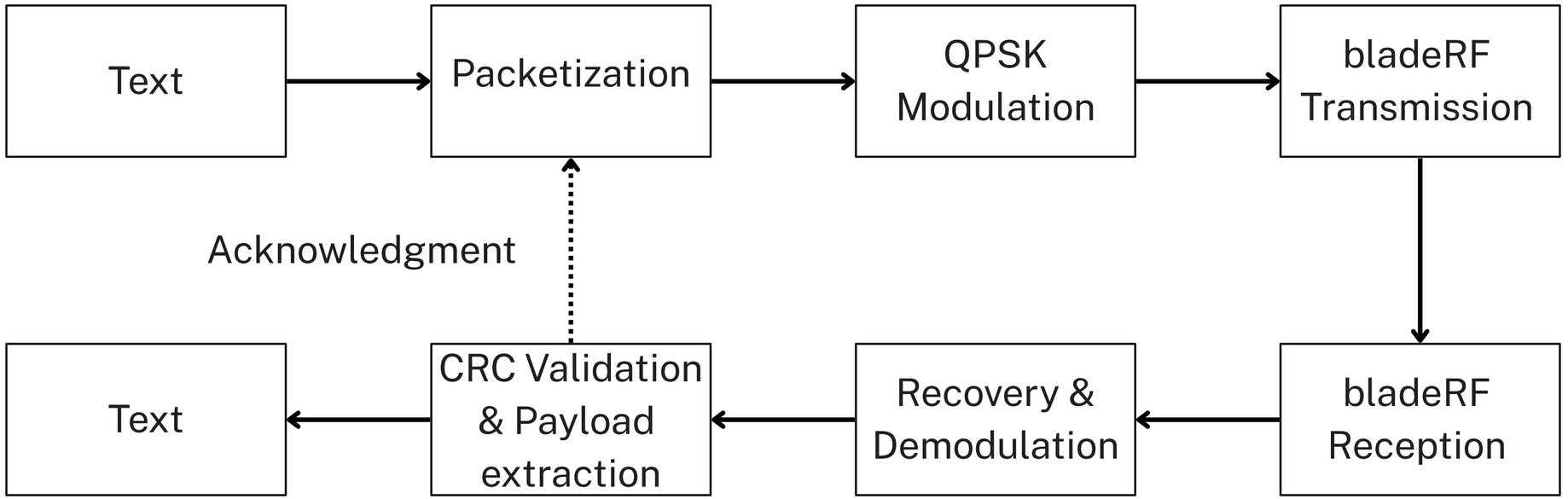
BY SPECACK

230147L – DILHARA D S
230508V – RAHUL B
230585C – SARUKA U
230654M – UMAIR A

Project Requirements

- Short message delivery system using digital modulation (QPSK)
- Unique user addressing
- Acknowledgment (ACK) mechanism for reliable communication
- CRC-based error detection and rejection of corrupted packets
- Basic GUI for message composing

One-Way Functional Block Diagram



Structure of a Packet (PAYLOAD)

PREAMBLE (128 B)	DESTINATION ADDRESS (1B)	SEQUENCE NUMBER (1B)	NONCE (8B)	CIPHER TEXT (32B)	CRC-32 (4B)

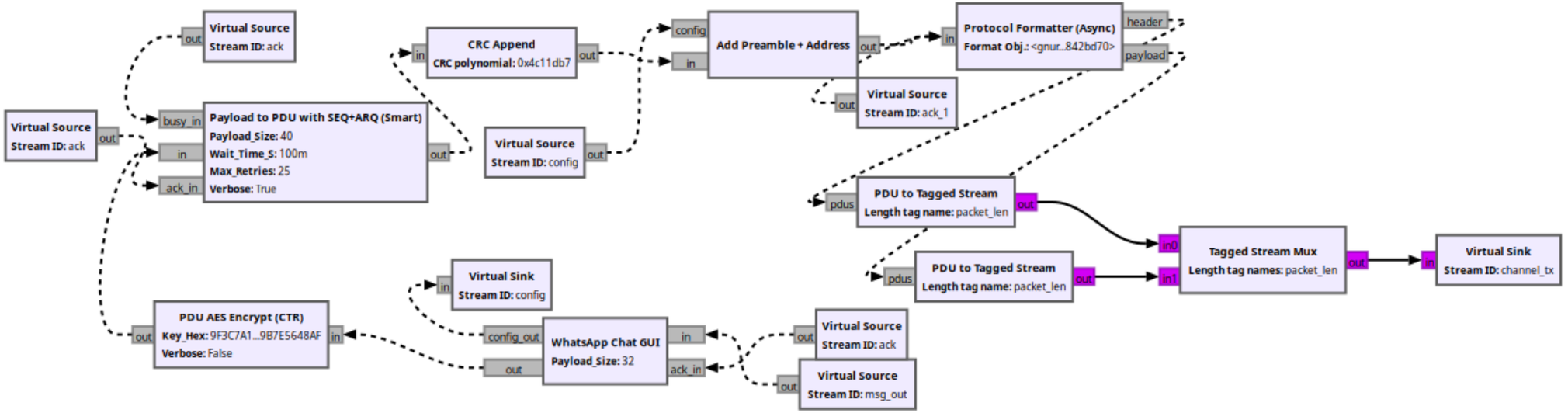
1. **Preamble:** Helps the receiver detect the start of a new packet, allows the SDR receiver to synchronize timing and carrier frequency before reading actual data
2. **Destination Address:** Indicates which specific device the packet is meant for and enables multi-node communication
3. **Sequence Number:** Provides a unique ID for each packet, and helps with tracking the acknowledgement part of the digital communication system.
4. **Nonce:** “Number used once”, for AES encryption and prevents replay attacks and ensures freshness of communication
5. **Cipher Text:** Contains the actual encrypted message from the user
6. **CRC-32:** Cyclic Redundancy Check value used to detect errors in incoming packets, computed over sequence number, nonce and the cipher text.

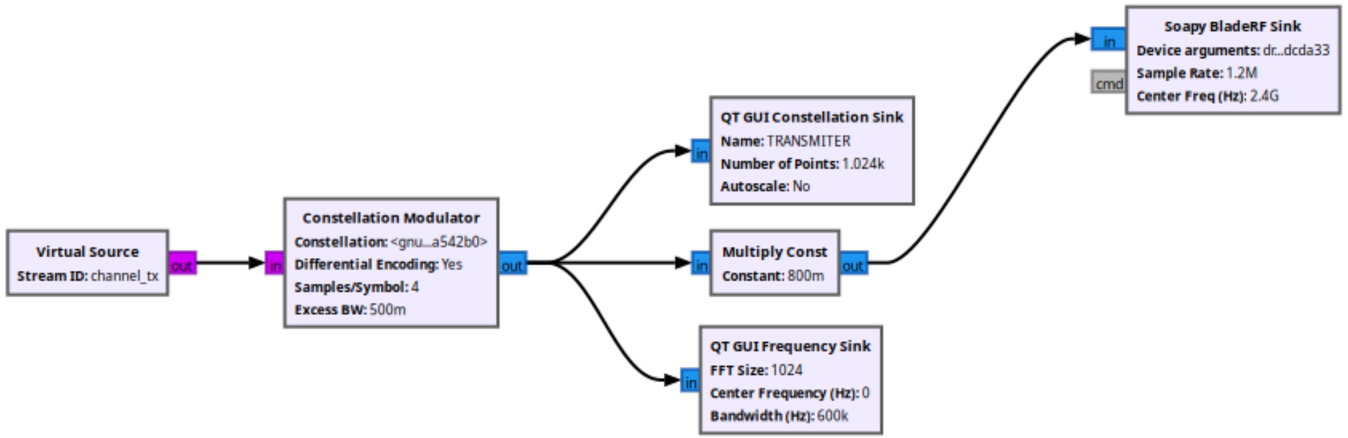
Structure of a Packet (ACK)

PREAMBLE (128 B)	DESTINATION ADDRESS (1B)	NEXT SEQUENCE NUMBER (1B)	PIGGYBACKED PAYLOAD (40B)	CRC-32 (4B)

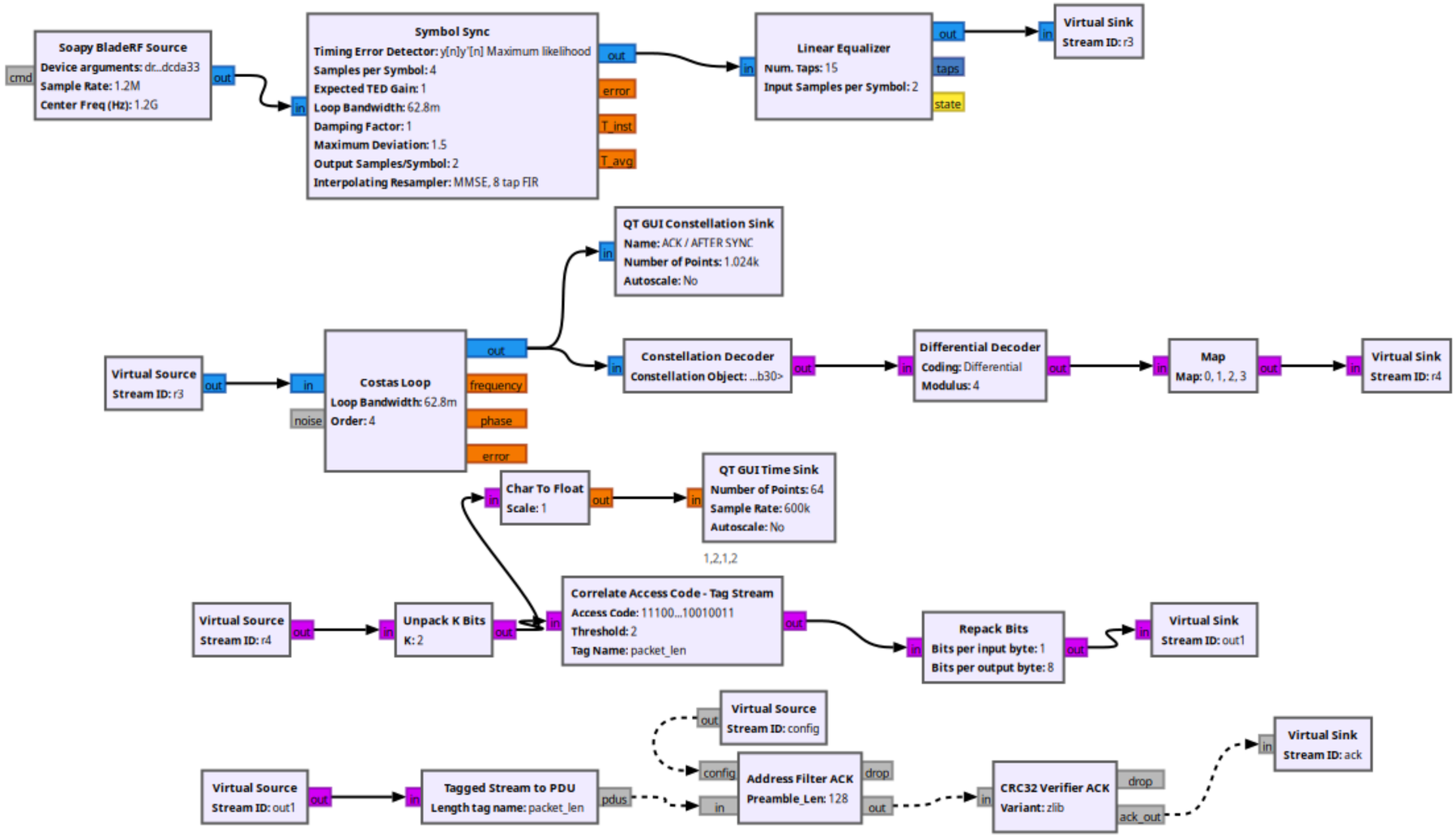
1. **Preamble:** Helps the receiver detect the start of a new packet, allows the SDR receiver to synchronize timing and carrier frequency before reading actual data
2. **Destination Address:** Indicates which specific device the acknowledgement packet is meant for and ensures the original sender processes the acknowledgement.
3. **Next Sequence Number:** Indicates the next expected sequence number from the sender and confirms that the previous packet was received successfully.
4. **Piggybacked Payload:** Allows the receiver to send extra information or a short return message together with the ACK
5. **CRC-32:** Ensures the ACK has not been corrupted during transmission.

Transmitter





ACK Receiver Block



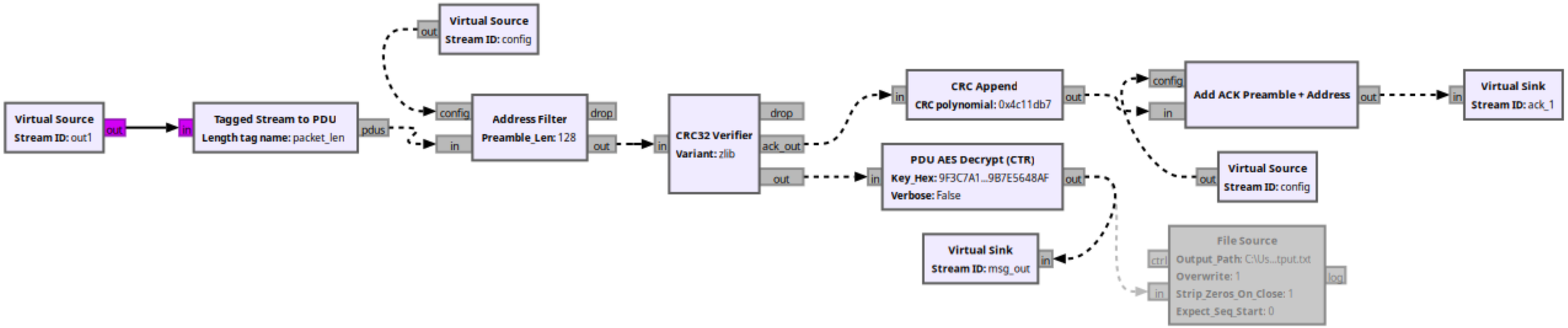
Functionalities of Transmitter Blocks

1. **WhatsApp GUI :** Custom Block which is a GUI with multi node transmission functionalities, allowing users to communicate with the desired user, also allows one user to receive and send messages at the same time.
2. **PDU AES Encrypt:** Takes the 32 byte PDU and encrypts the payload using AES in CTR mode and outputs a new PDU with nonce and ciphertext.
3. **Payload to PDU with SEQ+ARQ (Smart) :** Adds the sequence number to the packet for ARQ and also contains parameters for timeout period and maximum retries before moving onto the next packet, also prioritizes the ACK transmission before the payload transmission.
4. **CRC Append:** Appends CRC-32 to the end of the packet which is calculated using the payload and the sequence number.
5. **Add Preamble + Address:** Appends the destination address and the preamble to the packet
6. **Protocol Formatter:** Frames each packet with a header (containing sync word, addressing, etc.) and payload, ensuring proper synchronization and identification.
7. **PDU to Tagged Stream:** Converts packet-based data (PDUs) into tagged streams for modulation, and back again at the receiver. This allows GNU Radio blocks to handle both continuous streams and discrete packets.
8. **Constellation Modulator:** Maps digital bits into complex QPSK symbols for RF transmission.
9. **Symbol Sync:** Aligns the received signal samples with symbol timing to reduce inter-symbol interference
10. **Linear Equalizer:** Compensates for channel distortions and multipath effects, improving signal quality.

Functionalities of Transmitter Blocks

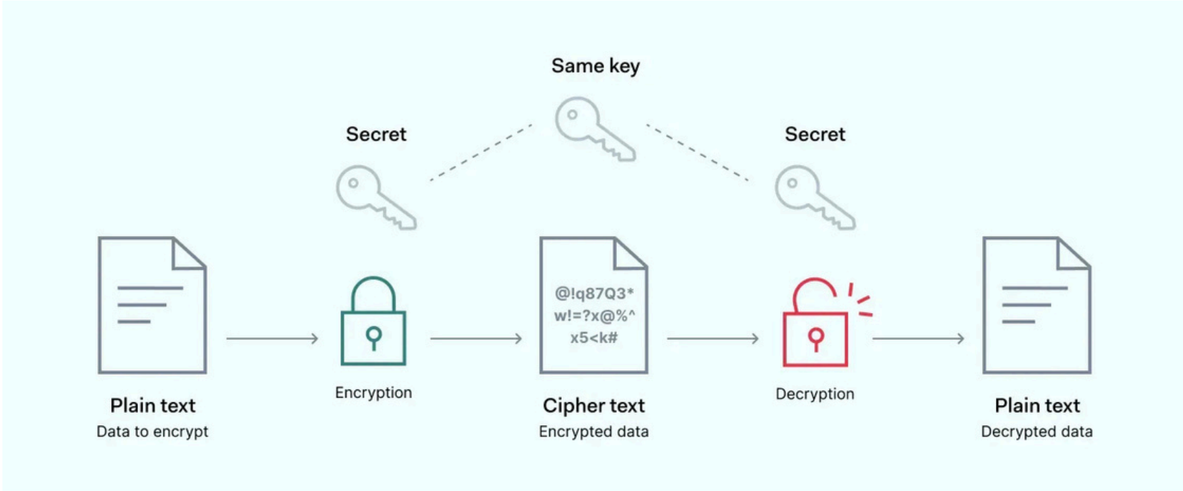
11. **Costas Loop:** Corrects carrier frequency and phase offsets in the received signal, enabling proper demodulation.
12. **Constellation Decoder:** Converts received QPSK symbols back into digital bits.
13. **Differential Decoder:** Removes phase ambiguity introduced during modulation/demodulation.
14. **Map, Unpack:** Reconstructs the bitstream into meaningful packets, preparing them for higher-layer processing
15. **Address Filter ACK:** Removes the preamble in the packet and checks if the address is correct and removes that as well forwarding the rest of the packet.
16. **CRC32 Verifier ACK:** Checks CRC-32 and forwards only the next sequence number if CRC check passes.

Message Receiver Block



Implemented Optional Features

1. **AES-CTR Encryption**
   - A symmetric key encryption algorithm used to secure digital data.
   - Operates on fixed-size blocks (128 bits) using 128, 192, or 256-bit keys.
   - Encrypts data by performing substitution, permutation, and mixing operations over multiple rounds.
   - CTR (Counter) mode allows encrypting variable-length messages and enables stream-like encryption.
   - Widely used in applications like secure messaging, file encryption, and network communications.

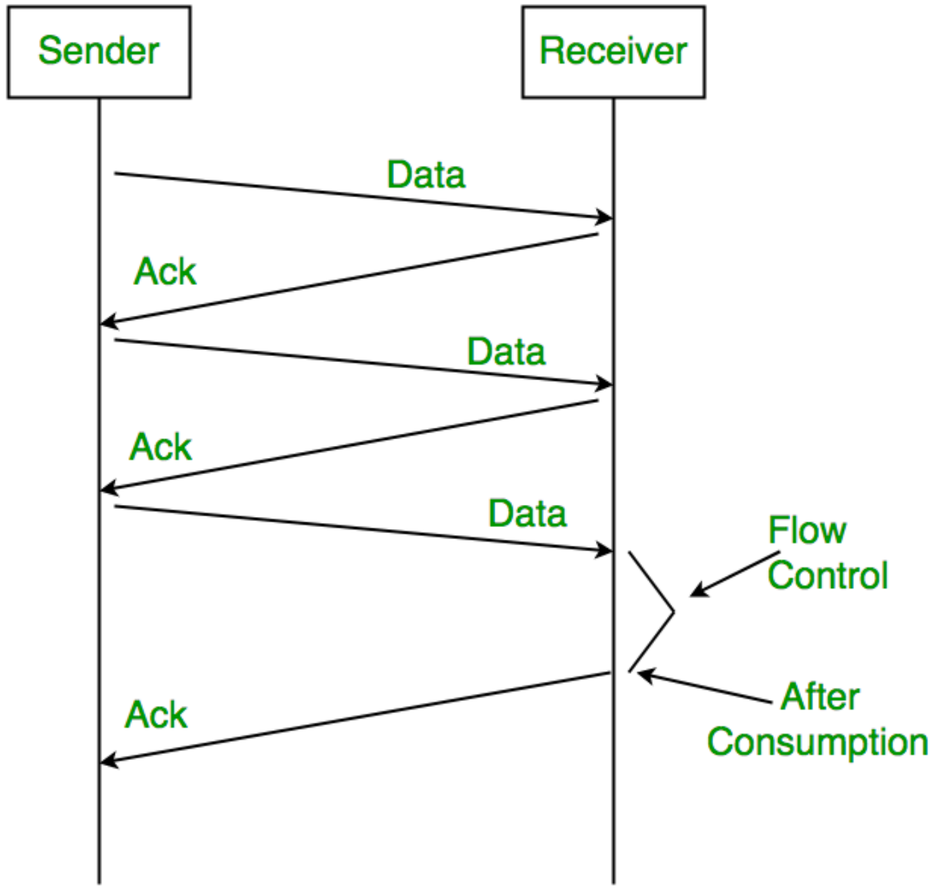


2. **GUI for Message Composing**
   - User-Friendly to easily compose messages similar to WhatsApp.
   - Provides the status of ACK by the use of colored ticks next to messages.
   - Easy to select the User it wants to communicate.
   - Can send text files

Selected Methodologies

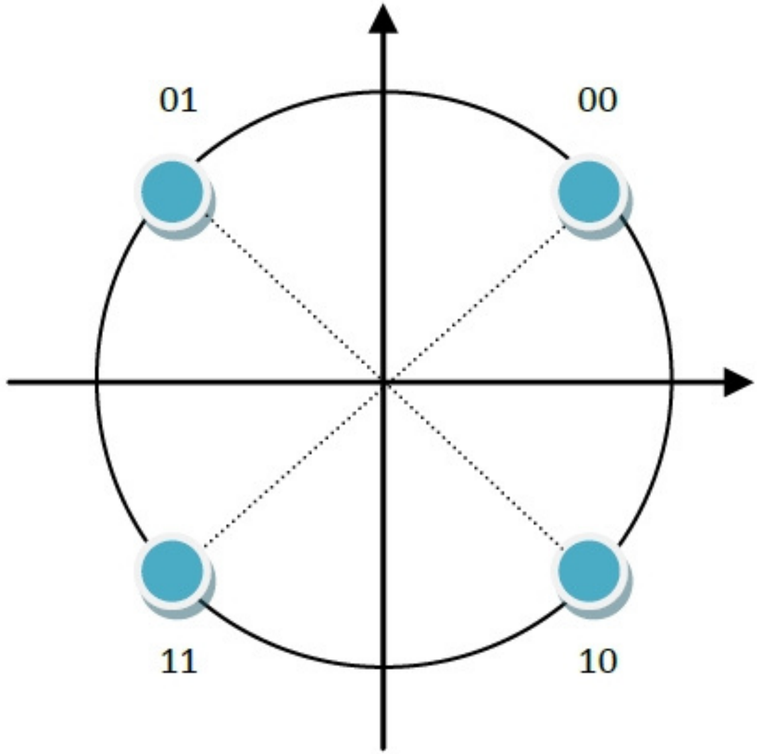
Stop-and-Wait ARQ

1. Each frame carries a destination address
2. The receiver checks for the correct address
3. If the address matches, the receiver sends an ACK back, acknowledging the successful packet transmission.
4. The sender waits looking for the ACK, if a certain timeout passes and still the ACK is nowhere to be found, the sender retransmits the frame assuming it got lost.

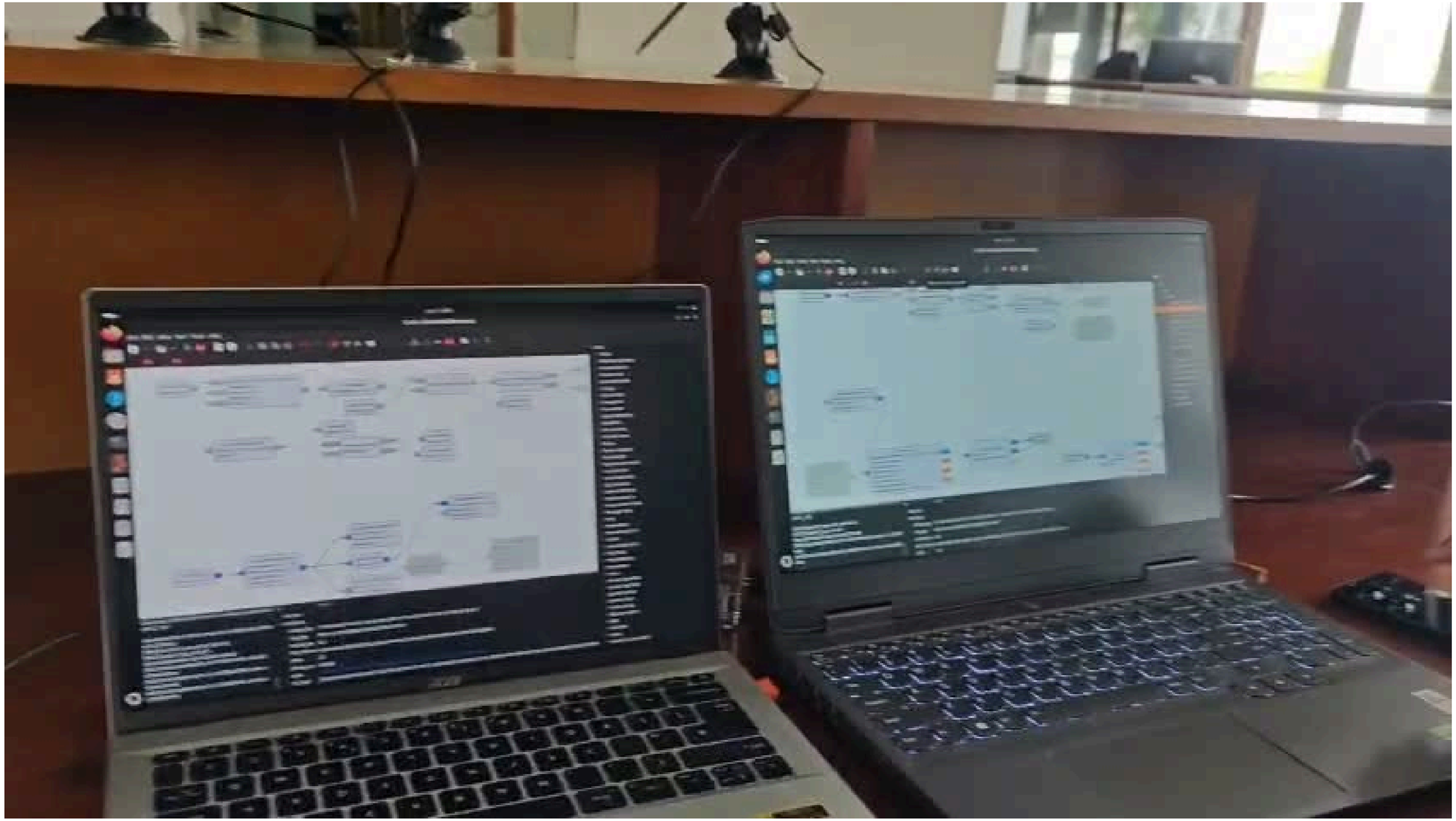


WHY QPSK?

1. **Better Data Rate with limited bandwidth**
   - QPSK carries 2 bits per symbol, doubling the data rate compared to BPSK while maintaining the same bandwidth.
   - For a messaging system, where throughput matters (ACKs), QPSK is very efficient.
2. **Tradeoff between Complexity and Performance**
   - Easier to implement in GNU Radio and on bladeRF while being simpler than other higher-order schemes.
3. **Strong Performance under noise**
   - Has excellent bit error rate in low or moderate SNR conditions.
   - Very ideal for lab environments and very robust.



FINAL TESTING



Thank you very much!