Information Transfer, using Ultrasound Waves

Application

Presented by: Yoni Grinberg & Bar Harush

Supervisor: Dr. Elena Ravve

Team Code:

What We Are Going To Do

Ultrasonic Data Transfer

Our project uses ultrasonic waves for data transfer, leveraging their ability to propagate through liquids and be contained by solid barriers.

Cross-Platform Network

A many-to-many ultrasonic data transfer network for Android and iOS, with advanced collision avoidance.

Why Ultrasound?

RF Challenges

- Crowded frequency spectrums.
- Susceptible to electromagnetic interference.
- Signal congestion and "snooping" risks.
- Poor propagation through conductive media (e.g., water, human body).

Ultrasound Advantages

- Propagates well through liquids and tissue.
- Naturally contained by solid barriers, enhancing security.
- Avoids crowded radio spectrums.
- Ideal for environments where RF fails (e.g., operating rooms, underwater).

Why Is The Project Not Trivial?

Use the same template in all slides: Here: title size

- **From PoC to Network Layer:** Transitioning from a basic one-to-one physical link to a robust, scalable **many-to-many** communication protocol.
- **Cross-Platform Interoperability:** Achieving seamless data exchange between heterogeneous mobile operating systems (**Android vs. iOS**) with unique audio chipsets and hardware constraints.
- **Physical and Algorithmic Constraints:** Managing severe acoustic challenges like **Multipath Propagation** (echoes) and **Self-Interference** on non-real-time mobile hardware.

Project Objectives

Bridging the Gap

From basic acoustic links to a robust, scalable, and secure network layer.

Secure Multi-User Network

Establish a secure, multi-user ultrasonic network for non-RF applications.

Advanced Arbitration

Implement CSMA/CA for sophisticated channel arbitration.

Cross-Platform Compatibility

Ensure seamless operation between Android and iOS devices.

Algorithmic challenges and solutions:

To small: not-readable. Divide into 2 slides

Mitigating Acoustic Interference and Noise

- **Challenge:** Indoor environments suffer from severe **Multipath Propagation**, where sound echoes overlap and cause Inter-Symbol Interference (ISI).
- **Solution:** Implementing a conservative **0.5ms bit duration** to allow signal echoes to decay and utilizing **Frequency-Shift Keying (FSK)** for robust signal modulation.

Decentralized Channel Arbitration

- **Challenge:** Managing many-to-many communication without a central server or global clock.
- **Solution:** Developing a **CSMA/CA-based protocol** using unique, **prime-number-based backoff timers** to minimize collision probability.

Overcoming the Hidden Node Problem

- **Challenge:** Undetectable collisions occurring when two transmitters, out of range of each other, simultaneously signal a central receiver.
- **Solution:** Utilizing a synchronized **RTS/CTS (Request to Send / Clear to Send) handshake** to logically reserve the medium at the receiver's end.

Ensuring Data Integrity in Noisy Channels

- **Challenge:** Environmental noise can easily flip bits in a 160-bit frame.
- **Solution:** Implementing a robust **16-bit CRC checksum** and an **ACK/NACK feedback loop** to verify and retransmit corrupted packets.

Building on Prior Work

Our project advances previous research in ultrasonic data transfer.

1

Foundational PoC

Previous teams established a one-to-one ultrasonic communication channel between two Android devices.

2

Physical Layer

Utilized Frequency-Shift Keying (FSK) for acoustic data transfer.

3

Frequency Selection

Determined 18-19 kHz as optimal for inaudible mobile communication.

4

Data Processing

Implemented Fast Fourier Transform (FFT) for decoding binary data.

Hardware Equipment

Heterogeneous Mobile Nodes (Receivers/Transmitters):

- We require at least **three concurrent devices** to validate the many-to-many network topology and collision avoidance.
- Testing must include a variety of models—specifically **two Android devices and one iPhone**—to account for different audio chipsets and microphone placements.

Cross-Platform Development and Compilation:

- An **Apple Mac** is an essential requirement for utilizing the iOS development environment and compiling code for the iPhone Operating System.
- Development relies on standard Android and iOS APIs, specifically **AudioTrack/AudioRecord** for Android and **AVAudioEngine** for iOS.

Testing Environment Tools:

- Access to standard physical obstacles (e.g., solid barriers and walls) is necessary to verify the **signal containment** and security properties of ultrasonic waves .
- Professional audio analysis software (e.g., **Audacity**) is required to verify that bit durations precisely meet the **0.5ms timing requirement** .

Addressing Previous Limitations

Past Challenges

- **Lack of Network Protocol:** Limited to one-to-one connections, unable to handle multiple devices simultaneously. Collisions occurred with three or more devices.
- **Single Platform Restriction:** Exclusively Android, limiting real-world application with heterogeneous devices (Android and iOS).

Our Impdrovements

- **Many-to-Many Communication:** Introducing CSMA/CA for sophisticated channel arbitration.
- **Cross-Platform Compatibility:** Ensuring seamless operation between Android and iOS.

What Is Our Approach? FSK algorithem

Explain abbriviation in the title not in the text

Protocol Overview & Parameters

- We utilize a multi-layered acoustic communication protocol designed for secure, many-to-many data transfer in non-RF environments.
- **Physical Layer:** Frequency-Shift Keying (FSK) modulation utilizing high-frequency carrier waves (18–20 kHz) to ensure inaudibility.
- **Data Link Layer:** A decentralized CSMA/CA arbitration mechanism with an RTS/CTS handshake to mitigate the Hidden Node Problem.
- **Protocol Inputs:**
  - **UserID** — 40-bit unique sender identifier.
  - **Sequence Number** — 8-bit packet order identifier.
  - **Payload** — 88-bit user message data.
- **Protocol Output:**
  - **Modulated Waveform** — FSK-encoded ultrasonic audio stream.

Algorithm (pseudo code)

```
Function TransmitSignal(bitStream):
track = InitAudioTrack(SAMPLE_RATE) for each bit in
bitStream:
  if bit == 0
     then targetFreq = 18000 HZ
     else targetFreq = 20000
    end if
     buffer = GenerateSineWave(targetFreq, phase)
ApplyAmplitudeRamp(buffer)
track.write(buffer)end for track.stop()End Function
```

What Is Our Approach? (CSMA/CA Protocol)

- **Mechanism:** We implement a decentralized, asynchronous "Listen-Before-Talk" principle to manage many-to-many communication without a central server.
- **Collision Avoidance:** Utilizing an RTS/CTS (Request to Send / Clear to Send) handshake to logically reserve the acoustic medium and solve the Hidden Node Problem.
- **Backoff Strategy:** We use unique prime-number-based timers to minimize the probability of simultaneous retransmissions.

Protocol Inputs:

- **RBWP** — Regular Base Waiting Period (high prime number).
- **MBWP** — Minimal Base Waiting Period (low prime number fallback).
- **Channel State** — Real-time detection of 18–20 kHz activity.

Protocol Outputs:

- **RTS/CTS Frames** — 56-bit control signals for medium reservation.
- **ACK/NACK** — Confirmation or error signals for frame integrity.

```
Function ExecuteCSMA(payload):

while True:

wait for RBWP_Timer to expire
if ChannelIsIdle(18-20kHz)
    then  Transmit(RTS_Frame)
    wait for CTS_Frame (Timeout = 2s)
    if CTS_Received
        then Transmit(Data_Frame)
         wait for ACK
         return Success
   else
       wait MBWP (Random Backoff)
   end if
else
    wait MBWP until idle
end if
end while
End Function
```

Why CSMA/CA Works

Collision Avoidance

CSMA/CA verifies channel availability before transmission, preventing signal collisions and packet loss.

Cross-Platform Compatibility

Robust frameworks exist for standardized Digital Signal Processing (DSP) and uniform hardware access.

Resilience to Dynamic Topology

Decentralized and asynchronous, allowing devices to join or leave instantly without network stalls.

Asynchronous Operation

"Listen-Before-Talk" principle is robust for devices with varying OS delays, unlike strict time synchronization.

Hidden Node Solution (RTS/CTS)

Explicitly reserves the medium at the receiver's end, ensuring all nodes know the channel is busy.

Explain abbriviation in the title not in the text

Implementation: Data Encoding (FSK)

Frequency-Shift Keying (FSK)

Our application uses FSK to encode binary data onto ultrasonic sound waves. This digital modulation technique changes the frequency of a carrier signal to transmit information.

- **Binary FSK (BFSK):** Carrier shifts between two discrete frequencies for binary (0s and 1s) information.
- **Frequency Selection:** 18 kHz for '0' and 20 kHz for '1', inaudible to humans.
- **Bit Duration:** Optimal 0.5 milliseconds per bit, allowing for signal stability and receiver sampling.

Implementation: Connection Establishment (RTS/CTS)

1. **RTS (Request to Send)**

   Transmitter sends a short control frame indicating intent to transmit and occupy the medium.

2. **Wait (SIFS)**

   Transmitter listens for a predefined window (Timeout).

3. **CTS (Clear to Send)**

   If the receiver is idle, it responds with a CTS frame, informing other participants the channel is busy.

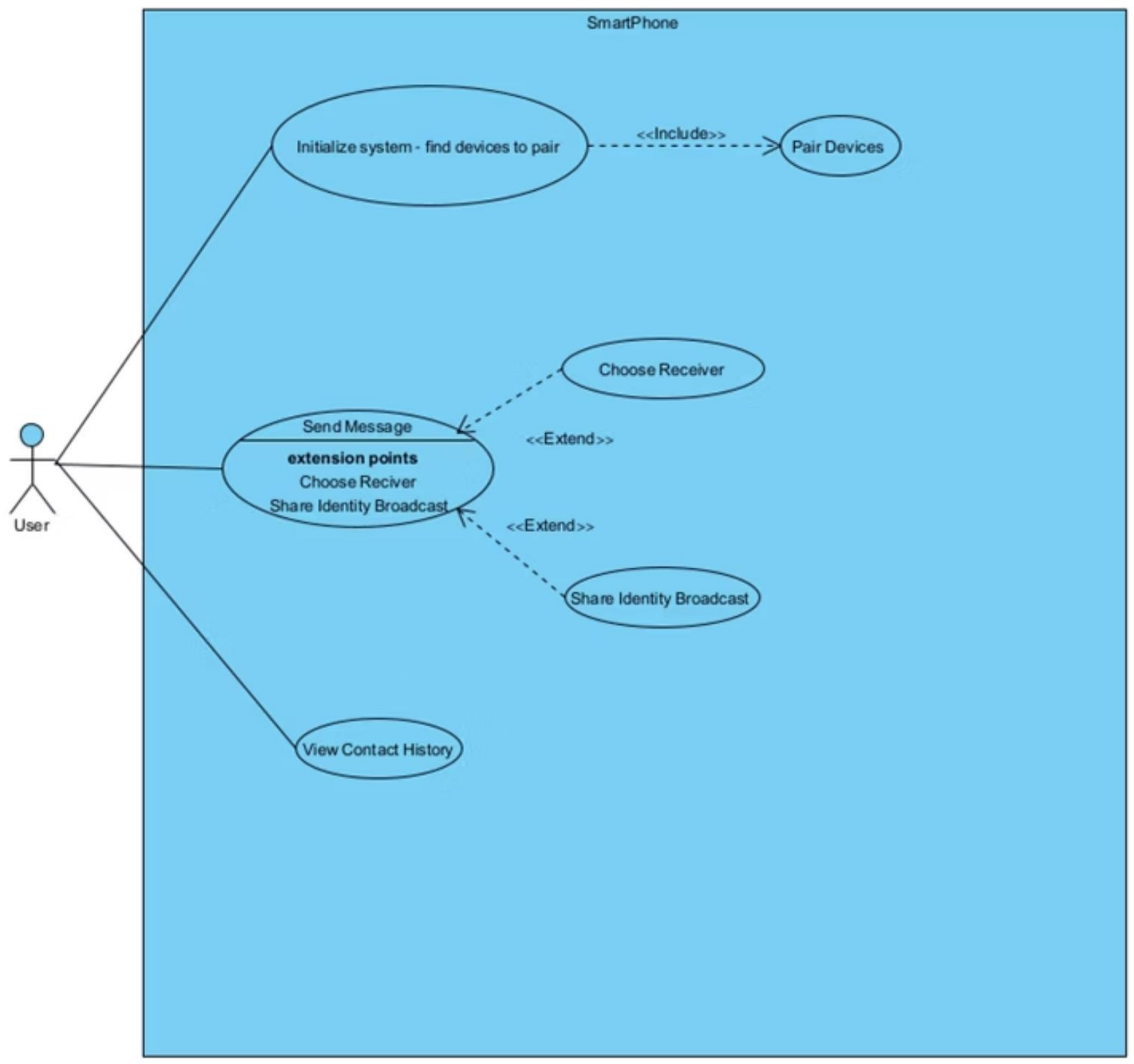
4. **Data Transmission**

   Upon receiving CTS, the transmitter sends the full data packet.

5. **Acknowledgement (ACK)**

   Receiver sends ACK upon correct data arrival; sender retransmits if no ACK.

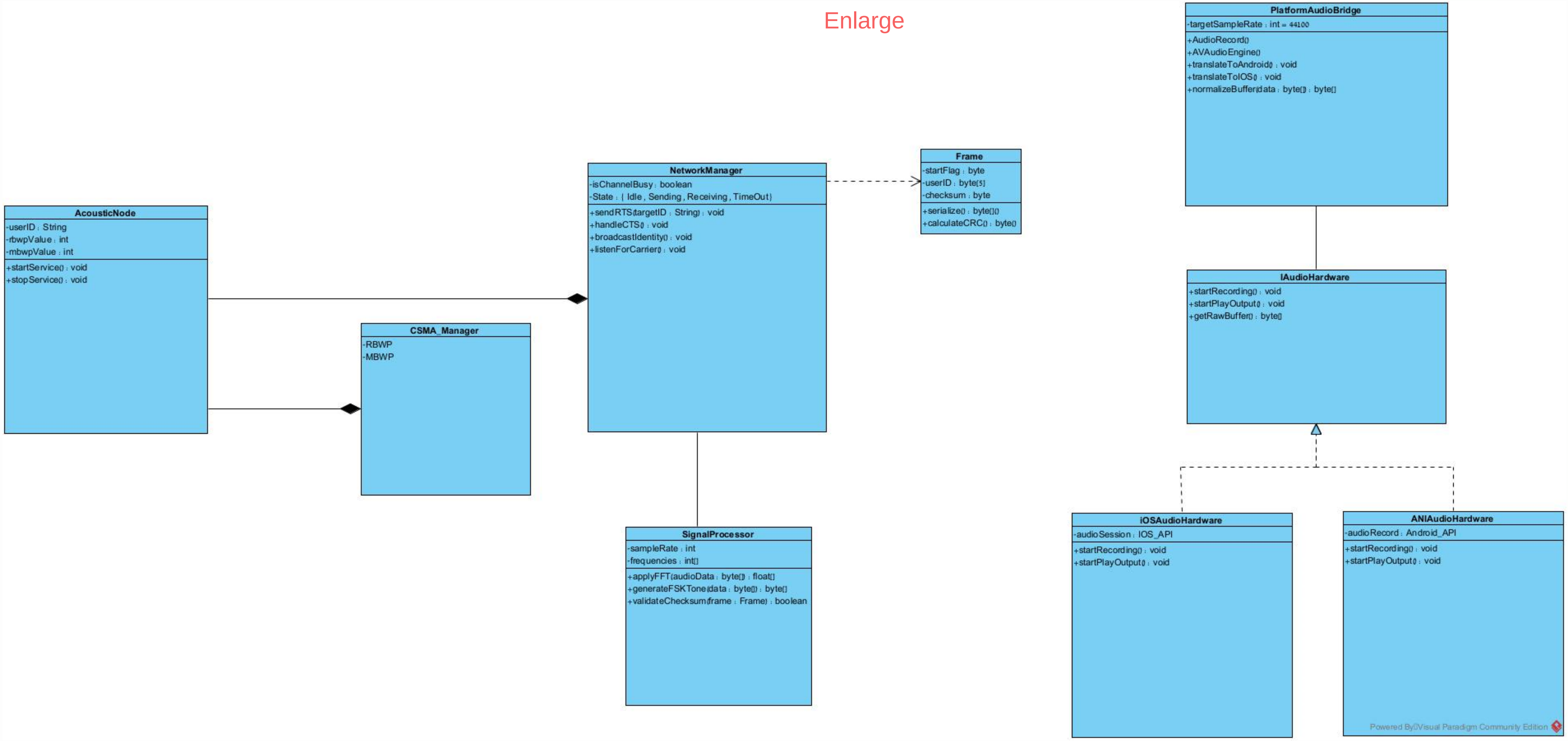
Use Case: Many-to-Many Interaction and Messaging Protocol



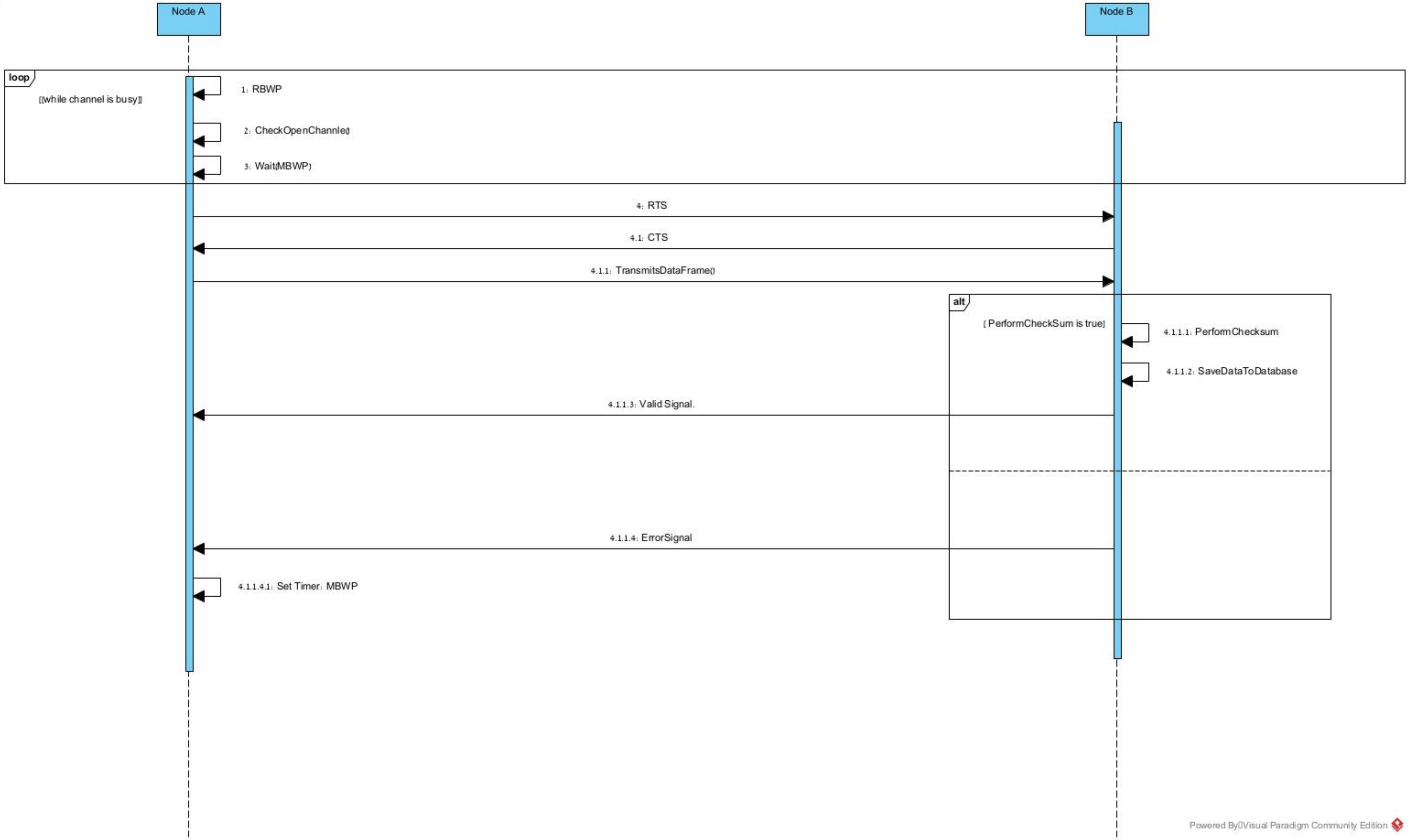
Enlarge

Class Diagram: Structural Design of the CSMA/CA and Signal Processing Layers

Enlarge



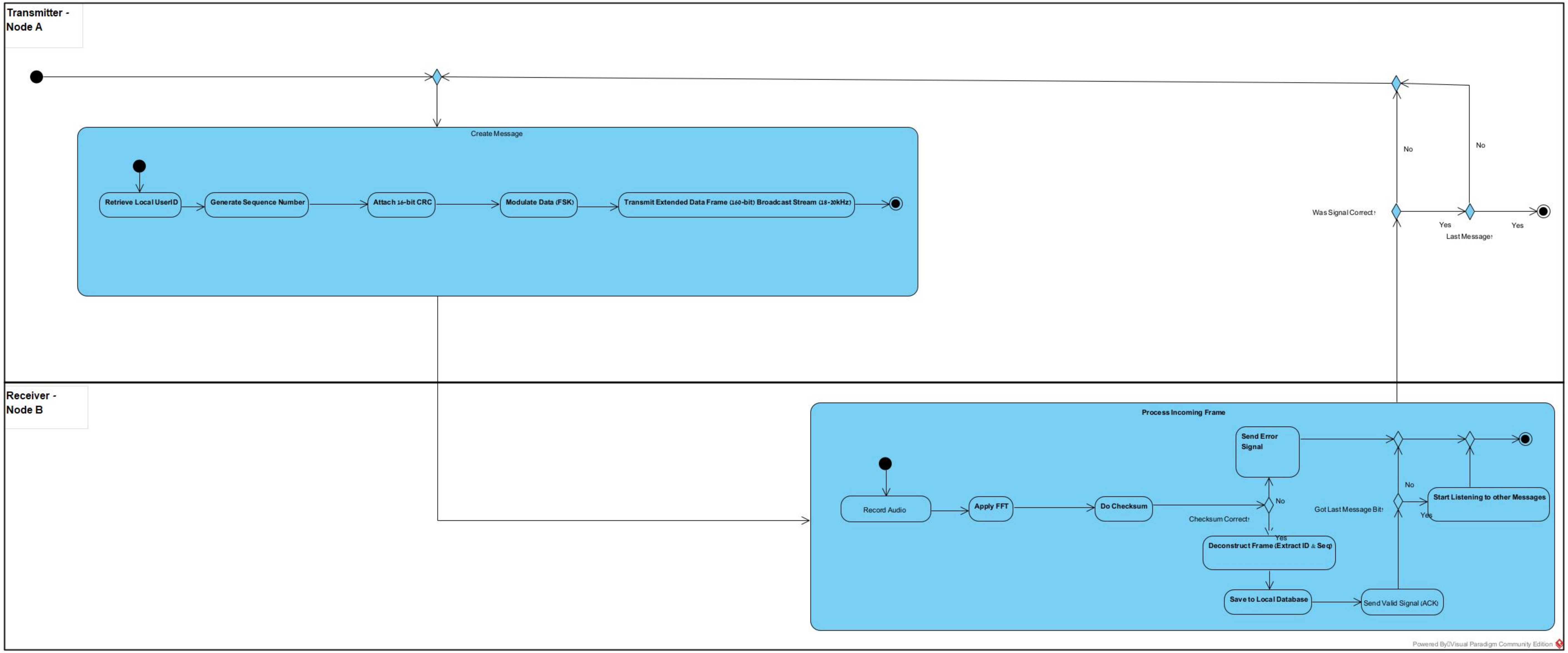
Sequence Diagram: CSMA/CA Protocol with RTS/CTS Handshake Sequence



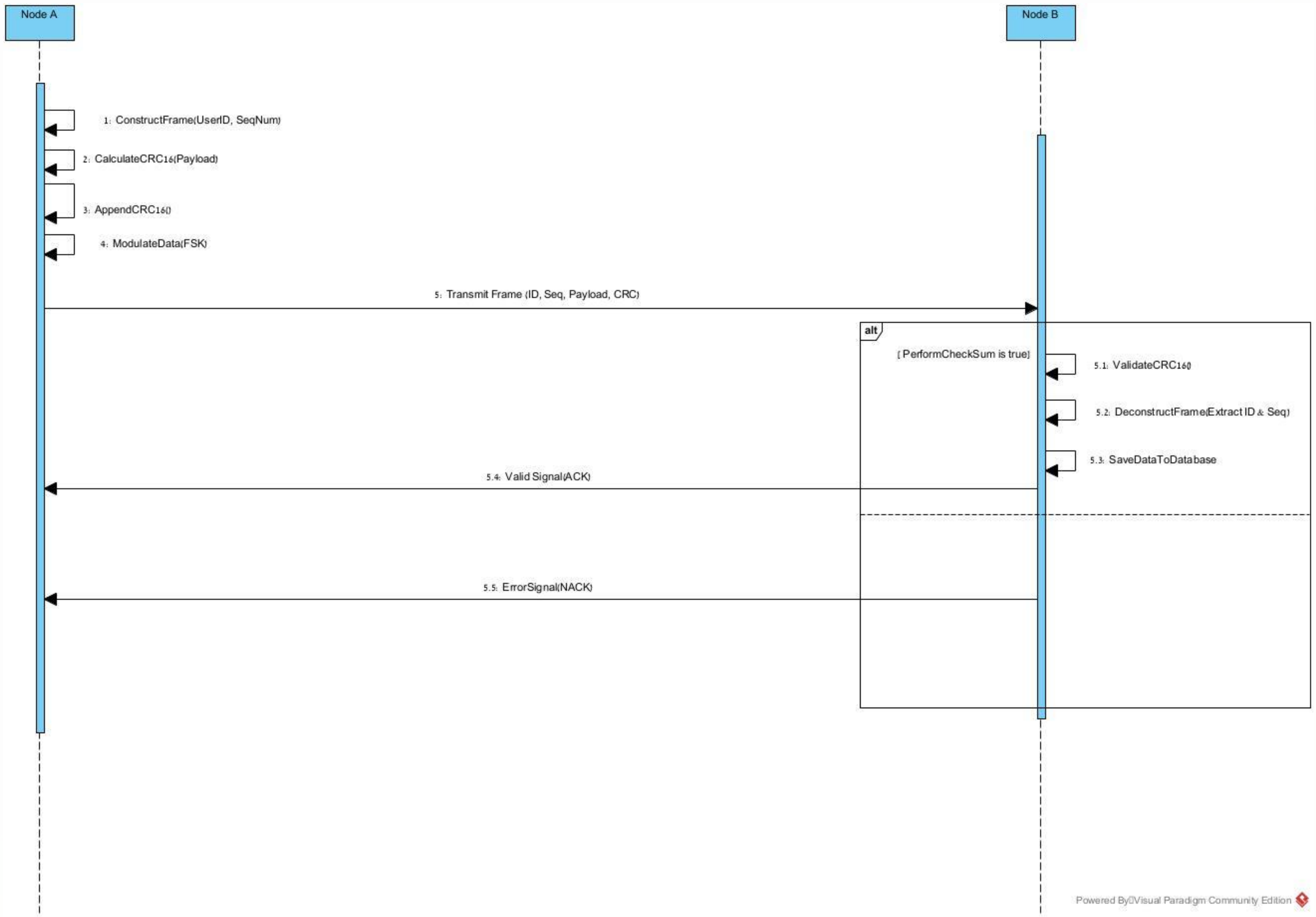
Enlarge

Activity Diagram: CSMA/CA Protocol with RTS/CTS Handshake

Enlarge



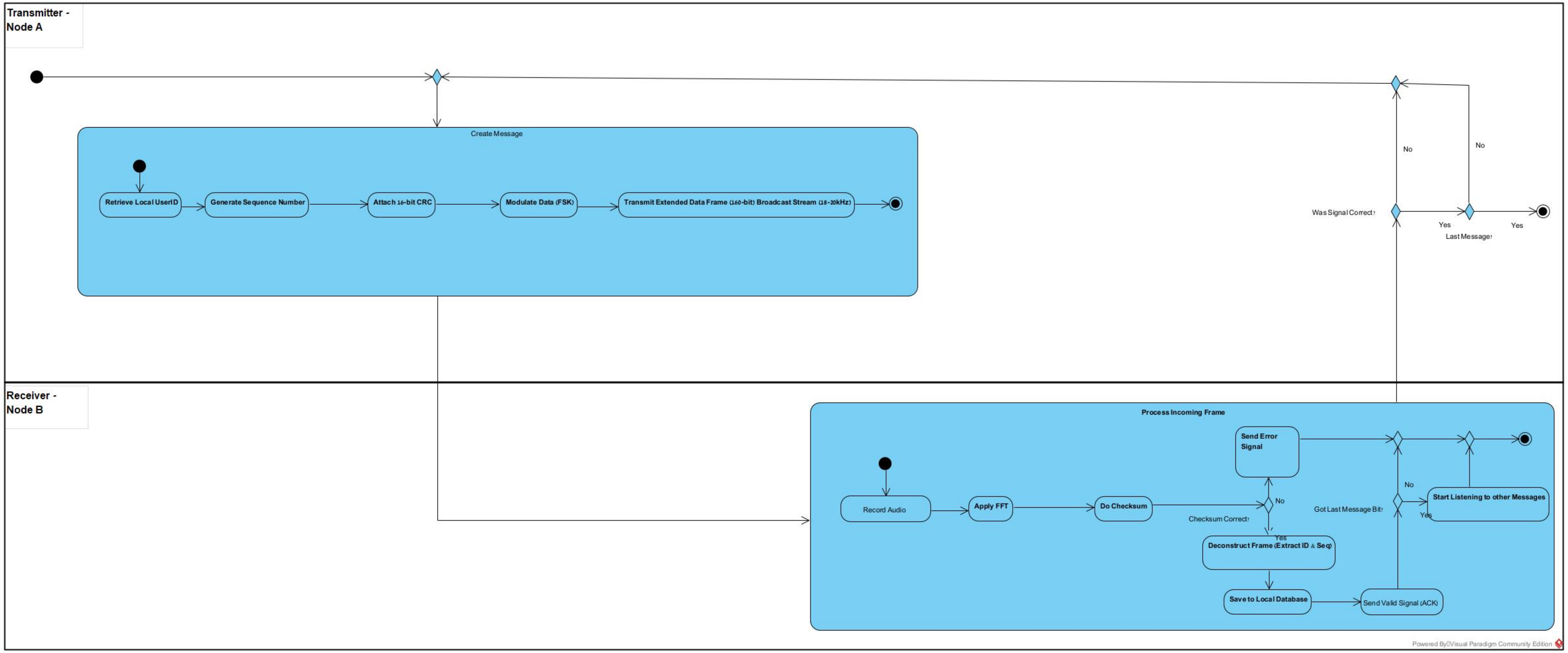
Sequence Diagram – Data Transmission and Frame Verification Protocol



Enlarge

Activity Diagram: Data Framing and Physical Layer

Enlarge



Expected Results & Key Specifications

Fully Functional Network

Acoustic communication network using ultrasonic waves.

Cross-Platform

Seamless communication between Android and iOS devices.

Reliable Protocol

Data Link Layer with RTS/CTS collision avoidance.

Capacity & Range

Supports 3+ concurrent nodes within a 2-meter radius.

Many-to-Many Topology

Manages data transfer from multiple emitters to multiple recipients.

Add a slide of testing

Add a slide of thanks