Implementation of Setup Phase in D2D Communication using SDR

1st Akshit Pal
2021EEB1150

2nd Anmol Yadav
2021EEB1153

3rd Harshal Sandhu
2021EEB1174

4th Parisha Aggarwal
2021EEB1049

***Abstract*—This paper presents the implementation of a setup phase in Device-to-Device (D2D) communication using Software-Defined Radio (SDR). The setup phase involves beacon signal transmission, device discovery, and data sharing between two devices. The communication is based on IEEE standard frame structures, with resource allocation for beacon signals and communication requests. Additionally, a selective mechanism is introduced to send communication requests only to nearby devices based on received signal power. Experimental results using the Adalm Pluto SDR demonstrate effective device discovery within a limited range, enhancing D2D communication efficiency.**

I. Introduction

Device-to-Device (D2D) communication has emerged as a promising technology for enabling direct communication between nearby devices without routing through base stations. This paradigm shift offers several advantages, including reduced latency, improved spectrum efficiency, and enhanced reliability, making it particularly appealing for applications such as proximity-based services, peer-to-peer networking, and Internet of Things (IoT) deployments.

This paper focuses on the implementation of the setup phase in D2D communication using Software-Defined Radio (SDR) platforms. SDR offers flexibility and programmability, allowing for rapid prototyping and experimentation with different communication protocols and algorithms. In particular, we investigate the use of the Adalm Pluto SDR, a low-cost, portable SDR platform with built-in RF capabilities, for implementing the setup phase of D2D communication.

The proposed approach leverages standard IEEE frame structures for beacon signal transmission and communication request handling. Additionally, we introduce a selective device discovery mechanism based on received signal power, which aims to target communication requests only to nearby devices, thereby reducing setup time and improving communication efficiency.

In this paper, we present the design and implementation of the setup phase using the Adalm Pluto SDR, along with experimental results demonstrating its effectiveness in device discovery and communication setup. We also discuss the implications of our findings and potential avenues.

II. Related Work

Overview of Device-to-Device (D2D) Communication Research
Device-to-Device (D2D) communication has emerged as a promising technology for enhancing communication efficiency and extending network coverage in wireless networks. Previous research efforts have explored various aspects of D2D communication, including protocol design, resource allocation, mobility management, and security mechanisms.

Research on D2D Communication Using Software-Defined Radio (SDR)
Researchers have leveraged Software-Defined Radio (SDR) platforms to conduct experiments and studies on D2D communication. SDR platforms, such as the Analog Devices ADALM-PLUTO, offer programmable radio capabilities, allowing researchers to prototype and evaluate D2D communication protocols and algorithms in real-world scenarios.

III. Methodology

The implementation of the setup phase consists of several key steps:

- Selective Device Discovery using Beacon
- Communication Request Handling
- Data Exchange

A. Selective Device Discovery using Beacon

Selective device discovery is a crucial aspect of the setup phase in D2D communication, as it determines which devices are targeted for further communication based on their proximity. In our implementation, we employ a selective device discovery mechanism based on received signal power.

The process begins with the transmission of beacon signals by devices. These beacon signals contain information about the transmitting device, including its sender's address (BSSID), which uniquely identifies the device. Nearby devices within the communication range receive these beacon signals, continuously monitoring the wireless channel for incoming transmissions.

Upon receiving a beacon signal, a nearby device measures the signal power of the received signal using metrics such as Received Signal Strength Indicator (RSSI). The measured signal power is then compared against a predefined threshold value.

If the received signal power exceeds the threshold value, indicating that the transmitting device is within proximity, the receiving device proceeds with further communication request handling. This may involve sending communication request frames, such as Request to Send (RTS) frames or similar messages, to initiate communication with the transmitting device.

On the other hand, if the received signal power is below the threshold value, suggesting that the transmitting device is distant or out of range, the receiving device ignores the beacon signal and does not send any further response. This selective approach ensures that communication resources are effectively utilized, directing communication efforts towards nearby devices and avoiding wasted resources on distant devices.

B. Communication Request Handling

After detecting beacon signals and confirming that the received signal power is above the threshold value, the receiving device initiates communication request handling. This process involves sending a signal similar to Request to Send (RTS) frames used in IEEE 802.11 Wi-Fi networks to the specific device identified by the received beacon signal.

The purpose of the RTS-like signal is to indicate to the target device that the sending device has information or data that it wishes to share. This serves as a preliminary step before the actual data exchange.

Upon receiving the RTS-like signal, the target device responds with a frame similar to Clear to Send (CTS) frames used in IEEE 802.11 Wi-Fi networks. The CTS-like frame indicates that the target device is ready to receive the data transmitted by the sending device.

It is important to note that while the target device typically responds with a CTS-like signal, there may be scenarios where it chooses not to send a response. This decision could be based on factors such as excessive data overload, security considerations, or other operational constraints.

The communication request handling process ensures that there is a mutual agreement between the sending and receiving devices before initiating data exchange. This helps prevent data collisions, improves communication reliability, and enhances overall system efficiency in D2D communication scenarios.

C. Data Exchange

After the device with information to share receives a Clear to Send (CTS)-like signal from another device, it proceeds with data transmission. This process involves configuring a data frame with a specified pattern and three addresses: receiver address, sender address, and BSSID (sender's address in our case). The payload of the data frame is created by converting the message or information to be shared into hexadecimal or octet format. This payload is then added to the payload section of the frame. Once the data frame is configured, it is transmitted by converting it into electromagnetic waves, similar to other signals, before being sent over the wireless channel. At the receiver's end, the transmitted signal is detected at the specified channel and decoded. The receiver checks if the frame type is data and verifies if the receiver's address matches its own address. If these conditions are satisfied, the payload or message signal is detected and processed by the receiver. Following this the device which received the message signal or data may send a pre-specified format acknowledge frame to the other end which will finally mark the end of whole process. In this way, a basic setup and data sharing are achieved in D2D communication scenarios. The process ensures that data is exchanged between devices efficiently and reliably, laying the foundation for further communication and collaboration in D2D networks.

IV. Code Flow

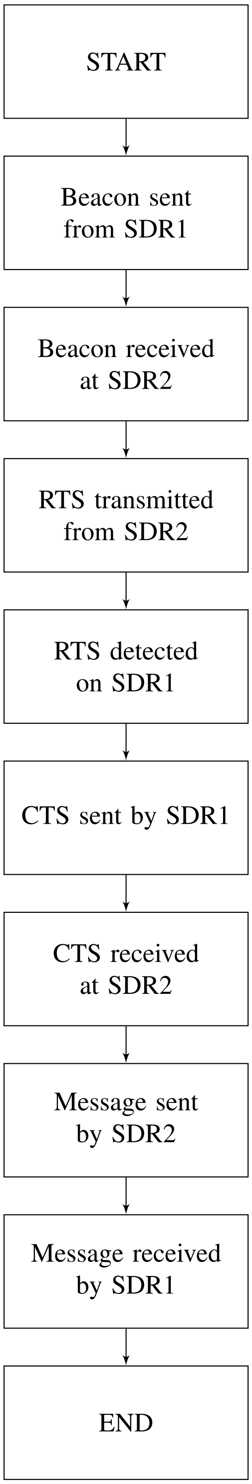
A. Codeflow of Device Transmitting Beacon

1) Set up the parameters for the SDR transmission and saving to file.
2) Configure the beacon frame with SSID, beacon interval, channel number, and other parameters.
3) Generate the beacon frame and convert it to waveform for transmission.
4) If saveToFile is true, save the waveform to a file.
5) If useSDR is true, configure the SDR parameters for transmission.
6) Transmit the beacon frame using SDR.
7) Receive the RTS signal using an SDR device or import it from a file.
8) Configure the SDR receiver parameters for channel scanning.
9) Scan and decode each channel for beacon signals.
10) Process the received beacon signals and extract information like SSID, BSSID, vendor, etc.
11) Display additional information about each successfully decoded beacon.
12) Determine the operating mode and channel width of the AP using the beacon information.
13) Generate a table of detected RTS information.
14) Release the SDR receiver object if used for reception.

B. Code Flow of Device Receiving Beacon

1) Set up parameters for receiving waveform.
2) Determine valid 20 MHz control channels.
3) Scan 5 GHz channels and configure SDR receiver.
4) Begin capturing and processing received packets.
5) Extract information from beacons.
6) Display information in a table format.
7) Send RTS signals to detected devices based on threshold power.
8) Wait for Clear to Send (CTS) signals.
9) Process received CTS signals and transmit message signals.

V. Transmitter and Receiver Flow Chart



VI. Experimental Setup

The experimental setup involves configuring the Adalm Pluto SDR devices for beacon signal transmission and reception. Two Adalm Pluto devices are placed within close proximity for device discovery and communication testing.

VII. Discussion and Future Improvements

A. Operation in Full duplex Mode

In our current setup, Software Defined Radios (SDRs) operate sequentially. One SDR sends a beacon frame, then waits for a Request to Send (RTS) and a subsequent Clear to Send (CTS) before proceeding with data transmission. However, transitioning to full-duplex mode allows SDRs to transmit and receive simultaneously, eliminating these waiting periods. With full-duplex capability, SDRs can immediately listen for RTS frames after sending the beacon and respond promptly, reducing idle time. During the RTS-CTS exchange, both SDRs can operate concurrently, significantly reducing latency. Once the CTS is received, data transmission commences without delay, optimizing communication efficiency and responsiveness. This transition maximizes resource utilization, enhancing overall system performance and scalability, thus representing a significant advancement in our setup.

B. Hard and Soft Handoff

Currently, our Software Defined Radios (SDRs) employ a fixed-handoff approach during communication, which can lead to inefficiencies and potential disruptions as devices move between coverage areas or encounter varying signal strengths. However, considering the adoption of both hard and soft handoff mechanisms represents a significant improvement in the future of SDR communication.

Combining both hard and soft handoff mechanisms offers a comprehensive solution that addresses the diverse requirements of different communication scenarios. By intelligently selecting between hard and soft handoff strategies based on factors such as device mobility, signal quality, and network congestion, SDRs can optimize handover performance while minimizing disruptions. Implementing a hybrid approach requires careful integration of hard and soft handoff protocols within the SDR communication stack, along with adaptive algorithms for dynamic handover decision-making. This approach ensures flexibility and responsiveness in handling handover events, ultimately enhancing the reliability and quality of communication services.

VIII. Results

```
Beacon sent and signal power 1.041867e-03.
Scanning channel 40 on band 5.0.
```

Fig. 1. Beacon sent

Fields	SSID	BSSID	Vendor	SNR_dB	Offset	Power
1	"TEST_BEACON1"	"0005F78A687F"	"Analog Devices, Inc."	31.8575	189170	2.6303e-06
2	"TEST_BEACON2"	"0005F78A777F"	"Analog Devices, Inc."	9.2866	1115035	1.1391e-06

Fig. 2. Beacon received

```
## Establishing connection to hardware. This process can take several seconds.
## Waveform transmission has started successfully and will repeat indefinitely.
## Call the release method to stop the transmission.
 RTS type Signal Sent 0005F78A687F.
```

Fig. 3. RTS sent

```
Signal Power 1.122324e-06.
Scanning channel 52 on band 5.0.
Signal Power 1.148674e-06.
Scanning channel 56 on band 5.0.
Signal Power 1.302807e-06.
Payload Size: 20 | Modulation: BPSK | Code Rate: 1/2
Type: Control | Sub-Type: RTSdefault SSID beacon detected on channel 56 in band 5.0.
sending cts 1## Establishing connection to hardware. This process can take several seconds.
```

Fig. 4. RTS received and CTS sent

```
Scanning channel 44 on band 5.0.
## Establishing connection to hardware. This process can take several seconds.
Signal Power 1.158819e-06.
Scanning channel 52 on band 5.0.
Signal Power 1.149536e-06.
Payload Size: 14 | Modulation: BPSK | Code Rate: 1/2
Type: Control | Sub-Type: CTS
 cts detected on channel 52 in band 5.0.
Scanning channel 100 on band 5.0.
Signal Power 1.141326e-06.
```

Fig. 5. CTS received

```
686921205468697320697320646576696365520312E
hi! This is device 1.
   5.2800e+09
## Establishing connection to hardware. This process can take several seconds.
## Waveform transmission has started successfully and will repeat indefinitely.
## Call the release method to stop the transmission.
 Message Signal Sent 0005F78A687F.
```

Fig. 6. Message sent

```
Signal Power 1.114350e-06.
Scanning channel 52 on band 5.0.
Signal Power 1.141524e-06.
Scanning channel 56 on band 5.0.
Signal Power 1.402769e-06.
Payload Size: 49 | Modulation: BPSK | Code Rate: 1/2
Type: Data | Sub-Type: Datadefault SSID data detected on channel 56 in band 5.0.
686921205468697320697320646576696365520312E
hi! This is device 1.
Scanning channel 100 on band 5.0.
Signal Power 1.155722e-06.
Scanning channel 100 on band 5.0.
Signal Power 1.155047e-06.
```

Fig. 7. Message received

IX. Conclusion

In this term project, we implemented the setup phase of Device-to-Device (D2D) communication using Software-Defined Radio (SDR), focusing on the Adalm Pluto SDR platform. Our approach incorporated selective device discovery and communication request handling, enabling efficient device discovery and communication setup within a limited range. By leveraging standard IEEE frame structures and a selective mechanism based on received signal power, we streamlined the setup phase of D2D communication. Experimental results validated the effectiveness of our methodology, demonstrating successful device discovery and communication initiation between nearby devices. Our project's practical implications include enhancing the efficiency and reliability of D2D communication systems for applications like proximity-based services and IoT deployments. Future work may involve further optimization of the device discovery mechanism and scalability considerations for larger deployments. Overall, our project contributes to advancing D2D communication technology and lays a foundation for future research in this area.

Acknowledgment

We wish to express our sincere appreciation to Professor Sam Darshi and his esteemed team of teaching assistants, Anandhu Ashok and Sagnik Bhattacharyya, for their invaluable support and guidance throughout our term project on SDR communication. Professor Darshi's profound expertise and mentorship provided us with invaluable insights and direction, enabling us to navigate the intricacies of the project effectively. Anandu and Sagnik's dedicated assistance, prompt feedback, and unwavering commitment to addressing our queries significantly enriched our learning experience and facilitated the project's progress.

We are deeply grateful for the conducive learning environment fostered by Professor Darshi and his team, which encouraged collaboration and mutual growth among students. Additionally, we extend our gratitude to our fellow group members for their exemplary teamwork and cooperation, which were pivotal to the project's successful completion

References

IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016). "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." IEEE Standard for Information Technology — Telecommunications and Information Exchange between Systems — Local and Metropolitan Area Networks — Specific Requirements.