COURSE DESIGN REPORT

Module 6: Simplex Control and Message/ACK Exchange

Renato Iida, Le Wang, Rebecca Cooper
Software Defined Radio System and Analysis

Abstract

In this project, we are going to create a simplex control mechanism to coordinate the communications between the nodes. First, we standardize the format of the frames. All the frames transmitted in the system have to be created based on the format. Then, we will utilize CRC-8 to verify the received frame and send the ACK frame if the verification is correct. Finally, we describe the steps and the interfaces with other groups to complete this project.

Problem Statement

In SDRSaA we are creating a Cellular Network with USRPs using Matlab. This network will be small, made up of only three users (UE 1, UE 2, and UE 3) connected to two base stations (BS 1 and BS 2). There will be three different standards used to communicate between nodes. As shown in Figure 1, UE 1 and UE 2 are associated with BS 1 and BS 1 connects to BS 2. BS 2 only has one client, UE 3. We are assuming all the communications between these nodes are on different channels. UE 1 talks to BS 1 through channel 1, UE2 talks to BS1 on channel 2. There will be at least 4 channels for the system. The USRPs run with Matlab cannot handle full duplex mode so each segment of communication will be in simplex mode. We are also assuming that the network will be static; none of the users will change base stations.

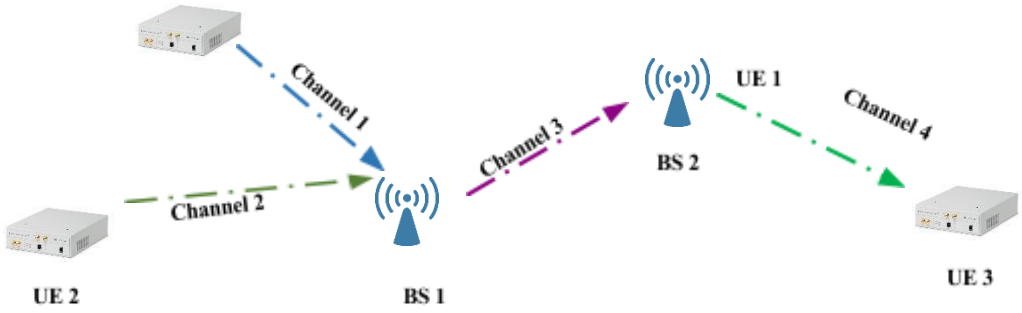


Figure 1: Overview of the cellular network architecture

As Team 6 we will be establishing end-to-end network resource allocation management as well as exchanging of control information and message/ACK forwarding between the UEs and the BS units. Other teams will be creating the three standards and the channels used to transmit messages as well as initializing the network and allocating the resources of each of the base stations.

Proposed Solution

To complete Module 6 and interface with the other teams we will need to create a simple MAC layer protocol, which should deal with routing protocol and checksum and create the initial messages that will be transmitted in the network.

Routing Protocol

There are two routing solutions that we came up with, static and dynamic, the one that we use will depend on the design choices of the other teams. The routing process will only be implemented by the base stations.

Static version

In this case, the frame leaves the sender, passes the base stations and reaches the destination receiver, through a static routing protocol. Based on the problem statement, we can draw static routing tables for both base stations. A draft of the routing tables is listed below:

Routing Table for BS1:

Src ID	Dst ID	Channel
UE 1	UE 2	Channel 2
UE 1	UE 3	Channel 3
UE 2	UE 1	Channel 1
UE 2	UE 3	Channel 3
UE 3	UE 1	Channel 1
UE 3	UE 2	Channel 2

Routing Table for BS2:

Src ID	Dst ID	Channel
UE 1	UE 3	Channel 4
UE 2	UE 3	Channel 4
UE 3	UE 1	Channel 3
UE 3	UE 2	Channel 3

For example, if UE 2 sends a frame to UE 3, first it will send the frame on channel 2, where only BS 1 can receive it. After receiving it, BS 1 will check the receiver address in the header, which appears as UE 3. Then BS 1 will send the frame on Channel 3. Similarly, only BS 2 can receive it. After checking the (Rcv ID) of the frame, BS 2 knows it is for UE 3, so BS 2 will send it on channel 4.

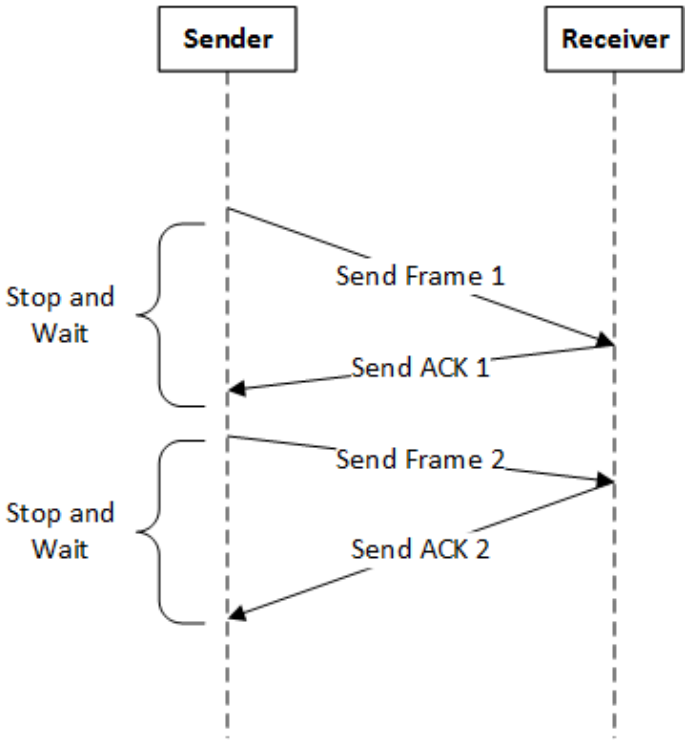
Dynamic Version

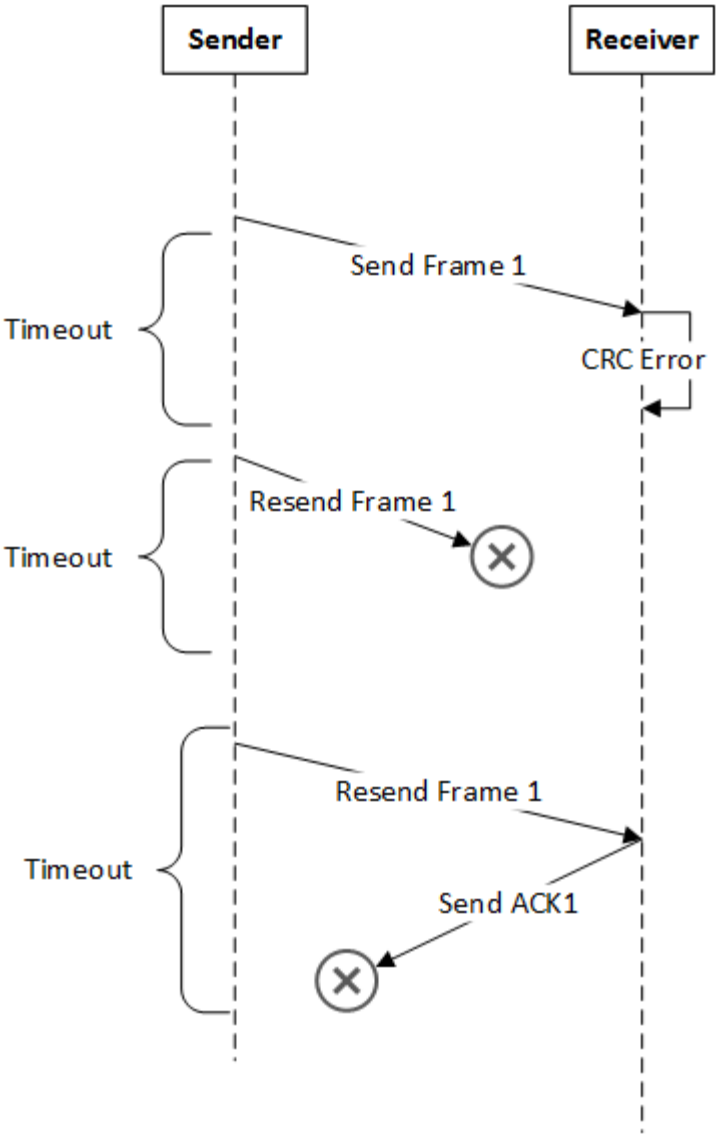
The dynamic algorithm is an open issue; the detail of this implementation will be discussed with team 4 and 5 in the development of this project.

Checksum Mechanism and ACK

We will build a checksum mechanism. Preliminarily, we adopt CRC-8 (Cyclic Redundancy Check). When a UE receives a data frame, it will perform CRC-8 on the payload. If the checksum returns a correct value, the receiver will reply with a control frame, ACK (acknowledge). CRC is simple and fast, however it does not have the ability to correct errors. The sender has to retransmit the frame if the CRC doesn't match instead of correcting locally. The whole system is in 'stop-and-wait' mode until the ACK is received by the UE.

'Stop-and-wait', also known as "positive acknowledgement with retransmission" is the fundamental technique to provide reliable transfer under unrealizable packet delivery system. In this mode, sender sends one data frame, waits for acknowledgement (ACK) from the receiver before proceeding to transmit next frame.





After broadcasting a frame, the sender will start a timer. If the sender cannot receive the ACK before timeout, it will retransmit. The reasons for such situation are listed below:

1. The frame reaches the receiver, however it is corrupted, which cannot pass the checksum mechanism. So no ACK will be replied, the sender will retransmit the frame after timeout.
2. The frame is lost during the transmission process. In other word, the receiver never received the frame. So there is no ACK, the sender will retransmit the frame after timeout.
3. The frame reaches the receiver and it is intact. The receiver did reply ACK, however this ACK frame is lost. The sender cannot receive ACK, so has to retransmit the frame after timeout.

In the worst case scenario there will be 6 transmissions between the initial transmission and the reception of the ACK by the UE. This is vital to determine how long to wait before the function will timeout and resend the frame.

Frame Format

When sending messages we will use at least two types of frames, the Data frame, and ACK frame. The frame type will distinguish between them. The data frame carries data and the ACK frame has no data payload. Additional types of frames can be added based on the needs of other teams.

The Rcv ID is the second part of the header so the base station only needs to check the first two bytes and it can make a decision on where to transfer the frame.

Based on the theory mentioned above, we design the format of frame as below:

Frame type	Rcv ID	Snd ID	SN	Data Size	CRC-8	Data (Payload)
1 byte	1 byte	1 byte	1 byte	2 bytes	1 byte	65,536 bytes

Rcv ID: The identification number of the receiver

Snd ID: The identification number of the sender.

SN: The sequence number is optional if it is necessary to deal with the situation when ACK is corrupt or lost.

Data size: Indicates the length of the payload.

CRC-8: The checksum result.

Data: The payload of the frame.

Implementation

We will create Matlab functions to complete our module's functionality.

- Function *Framing ( )*
    - Get the information such as Destination ID, Channel number from broadcast table.
    - Calculate the CRC of the data.
    - Create the header of the frame based on the information above and append the header to the payload to form the frame.
    - Pass the frame to the teams 1, 2 or 3 to send the data over the air.

- Function *Extract ( )*
    - Occurs after we receive the frame from teams 1, 2, or 3.
    - Extract the payload from it.
    - Call the Checksum function to calculate the CRC and check if it is correct.
    - Interface with Team 4 to send ACK.
- Function *Checksum ( )*
    - Calculate the CRC of the data and check it with the one from the header.
    - Return correct if they are equal.
- Function *Routing ( )*
    - Implement routing algorithm according to the routing table.

Prototype Evaluation Strategies & Logistics

- Milestone 1 (02/06):
  We will create a script in matlab that will create the message and it calls the function *Framing ( )*. This script will include placeholders for the functions that have not yet been developed from all the teams. The other milestones will replace them with functional code. It will make the integration processes more efficient and incremental with this team and the others. The *Framing ( )* function will be evaluated on whether it can create a frame of data from our placeholders. We will be working with the other teams to determine if our placeholders are appropriate.

- Milestone 2 (02/13):
  For this milestone we will add the verification of the received packet to the script developed in the Milestone 1. At this point we will have worked with team 4 to determine how the ACK will be sent and add that to our script.

- Milestone 3 (02/20):
  Here we will add the routing algorithm and interface to send and received the frame from team 1, 2, and 3. It will evaluate the end-to-end communication of the frame. The test case will follow the scenarios described in the Course design project listed below:

    • Extra-cellular communications between UEs:

        US1 -> BS1 -> BS2 -> US3

        US3 -> BS2 -> BS1 -> US1

        US2 -> BS1 -> BS2 -> US3

        US3 -> BS2 -> BS1 -> US2

    • Intra-cellular communications between UEs:

        US1 -> BS1 -> US2

        US2 -> BS1 -> US1