PDCP Report

Viknesh S
Ganesh Ayyappan

5G Testbed Summer Interns

Contents

1 Overview

- Our work majorly involved programming the protocols of the PDCP layer (part of layer 2) of 5G network.
- The following codes are part of the main task given. The codes are in the PDCP folder.
  - **PDCP_base.h**
  - **rohc_initial.h**
  - **t-reorder.c**
  - **server_srb.c**
  - **client_srb.c**
  - **server_drb1.c**
  - **client_drb1.c**
  - **server_drb2.c**
  - **client_drb2.c**

2 Code Explanations

2.1 PDCP_base.h

- Contains all the required structures and function prototypes along with definitions for various types of PDCP PDUs.
- Struct definitions of various PDCP PDUs

2.1.1 SRB Data PDU

- **R_SN**: 16-bit *unsigned int* variable in which first 4 bits (from the MSB side) represent R-bits (reserved bits) and remaining 12 bits represent Sequence Number.
- ***data**: 8-bit *unsigned int* pointer variable which points to the starting of the data.
- **data_len**: *int* variable which stores length of the data in bytes. Along with *data variable, whole of the data can be accessed as and when necessary.
- **set**: *boolean* variable which denotes whether header has been added or not. It is "true" if header has been added.
- **MAC**: 32-bit *unsigned int* variable. MAC (Message Authentication Code) is a special information required for Integrity protection.

2.1.2 DRB-1 Data PDU

- **DC_R_SN**: 16-bit *unsigned int* variable in which first bit (from the MSB side) denotes D/C bit (Data/Control PDU. If 1, it is a Data PDU; else Control PDU), next 3 bits represent R-bits and remaining 12 bits represent Sequence Number.
- ***data**: 8-bit *unsigned int* pointer variable which points to the starting of the data.
- **data_len**: *int* variable which stores length of the data in bytes. Along with *data variable, whole of the data can be accessed as and when necessary.
- **set**: *boolean* variable which denotes whether header has been added or not. It is "true" if header has been added.
- **MAC**: 32-bit *unsigned int* variable.

2.1.3 DRB-2 Data PDU

- **DC_R_SN**: 8-bit *unsigned int* variable in which first bit (from the MSB side) denotes D/C bit, next 5 bits represent R-bits and remaining 2 bits represent the first 2 bits from the MSB side of Sequence Number.

- **SN**: 16-bit *unsigned int* variable in which represents the remaining 16 bits of Sequence Number .
- ***data**: 8-bit *unsigned int* pointer variable which points to the starting of the data.
- **data_len**: *int* variable which stores length of the data in bytes. Along with *data variable, whole of the data can be accessed as and when necessary.
- **set**: *boolean* variable which denotes whether header has been added or not. It is "true" if header has been added.
- **MAC**: 32-bit *unsigned int* variable.

2.1.4 Status Control PDU

- **DC_Type_R**: 8-bit *unsigned int* variable in which first bit (from the MSB side) denotes D/C bit, next 3 bits represent PDU type (indicates the type of control information included in the corresponding PDCP Control PDU. 000 is for STATUS) and remaining 4 bits represent R-bits.
- **FMC**: 32-bit *unsigned int* variable which denotes "First Missing Count" of PDCP SDUs.
- ***BitMap**: 8-bit *unsigned int* pointer variable which points to the starting of the BitMap.
- **BitMap_len**: *int* variable which stores length of the BitMap in bytes. Along with *BitMap variable, whole of the data can be accessed as and when necessary.

2.1.5 ROHC Control PDU

- **DC_Type_R**: 8-bit *unsigned int* variable in which first bit (from the MSB side) denotes D/C bit, next 3 bits represent PDU type (001 for ROHC) and remaining 4 bits represent R-bits.
- ***ROHCfb**: 8-bit *unsigned int* pointer variable which points to the starting of the Interspersed ROHC feedback.
- **ROHCfb_len**: *int* variable which stores length of the Interspersed ROHC feedback in bytes. Along with *ROHCfb variable, whole of the data can be accessed as and when necessary.

- Roles of Functions

2.1.6 Initialization functions

- To give default values to all the variables mentioned above.
- Present for all PDUs

2.1.7 Data generation functions

- This is a temporary function to generate test data.
- This may be done away with once the code is put into real-time use.
- Present for all Data PDUs

2.1.8 Header Operation functions

- This function adds or removes header using bit manipulation techniques and "set" variable.
- Present for all Data PDUs

2.1.9 Integrity Protection functions

- This function is for integrity protection purposes. (MAC generation)
- As of now, only function prototypes are present.
- Present for all Data PDUs.

2.1.10 Ciphering functions

- This function is for ciphering and de-ciphering purposes.
- As of now, only function prototypes are present.
- Present for all Data PDUs.

2.2 rohc_initial.h

- ROHC stands for **RO**bust **H**eader **C**ompression
- Contains ROHC() and ROHD() functions which perform Compression and Decompression on IP Address respectively.
- These Compressor and decompressor functions are using default IP Address for testing. This may be done away with once the code is put into real-time use.
- The Compression and Decompression Algorithms are present in the ROHC libraries. For further reference, kindly visit *https://rohc-lib.org/*

2.3 t-reorder.c

- This code is the implementation of PDCP packet t-reordering algorithm on simpler test data.
- The implementation is done based on the flowchart provided to us. Kindly refer the t-Reordering Refernce folder in the PDCP folder for the same.
- This is further integrated with the original program. Thus, this code as an individual has no link to original one. This is just for reference.

2.4 server_*data_pdu*.c

- Description for **server_*data_pdu*.c** is applicable for **server_srb.c**, **server_drb1.c** and **server_drb2.c**.
- Connection is established between Server and corresponding Client program using TCP and sockets.
- The server codes perform the following:-
  - The whole packet is received from the **SDAP** layer.
  - ROHC() is called so as to compress the IP address of the packet received.
  - Integrity Protection and Ciphering are performed. (Only function prototypes available as of now)
  - PDCP packet of the corresponding data PDU is initialized with the memory being allocated dynamically.
  - In the code, data generation function is called. This should be replaced with the data received.
  - PDCP header is added.
  - The whole packet is transmitted to **client_*data_pdu*.c** using **send**().

2.5 client_*data_pdu*.c

- Description for **client_*data_pdu*.c** is applicable for **client_srb.c**, **client_drb1.c** and **client_drb2.c**.
- Connection is established between Client and corresponding Server program using TCP and sockets.
- The client codes perform the following:-
  - The whole packet is received from the **server_*data_pdu*.c** using **recv**().
  - PDCP header is removed.
  - Reordering of the received packets implemented using t-reordering algorithm. Some packets are discarded in this process.
  - Integrity Protection and Deciphering are performed. (Only function prototypes available as of now)
  - The whole packet is transmitted to the **RLC** layer.

THE END