TEST 4: PDCP LAYER – SECURITY AND HEADER COMPRESSION

REFERENCE STANDARD: 3GPP TS 38.323

LAYER: PDCP LAYER

DIFFICULTY LEVEL: ADVANCED

SUBMITTED BY:

GROUP 3 - TEAM MEMBERS:

- ***55984_Vikas Srivastava***
- ***58622_Harshinie M***
- ***58623_Shreyash Bhatt***
- ***58624_Jayavarshini G***
- ***58635_Yuvaraj P***

TABLE OF CONTENTS

1. INTRODUCTION TO PDCP LAYER

The Packet Data Convergence Protocol (PDCP) layer is an important layer in the 5G NR protocol stack that ensures efficient and secure transmission of data between the User Equipment (UE) and the network. It operates above the Radio Link Control (RLC) layer and supports both control plane and user plane communications.

The PDCP layer performs key functions such as header compression, ciphering, integrity protection, packet sequencing, and in-order delivery. These functions help reduce transmission overhead, improve bandwidth utilization, and maintain data security during communication.

In this project, the PDCP layer operations are simulated using a C++ implementation to demonstrate security processing and header compression mechanisms. The simulation helps in understanding how PDCP processes packets before transmission and how efficiency and reliability are maintained in a 5G environment.

3GPP Reference

The PDCP layer functionality used in this project is based on the standards defined by the 3rd Generation Partnership Project (3GPP). The specifications mainly refer to 3GPP TS 38.323, which defines PDCP procedures for 5G NR. Security-related operations follow guidelines from 3GPP TS 33.501. These standards ensure interoperability and standardized implementation of PDCP functions in 5G networks.

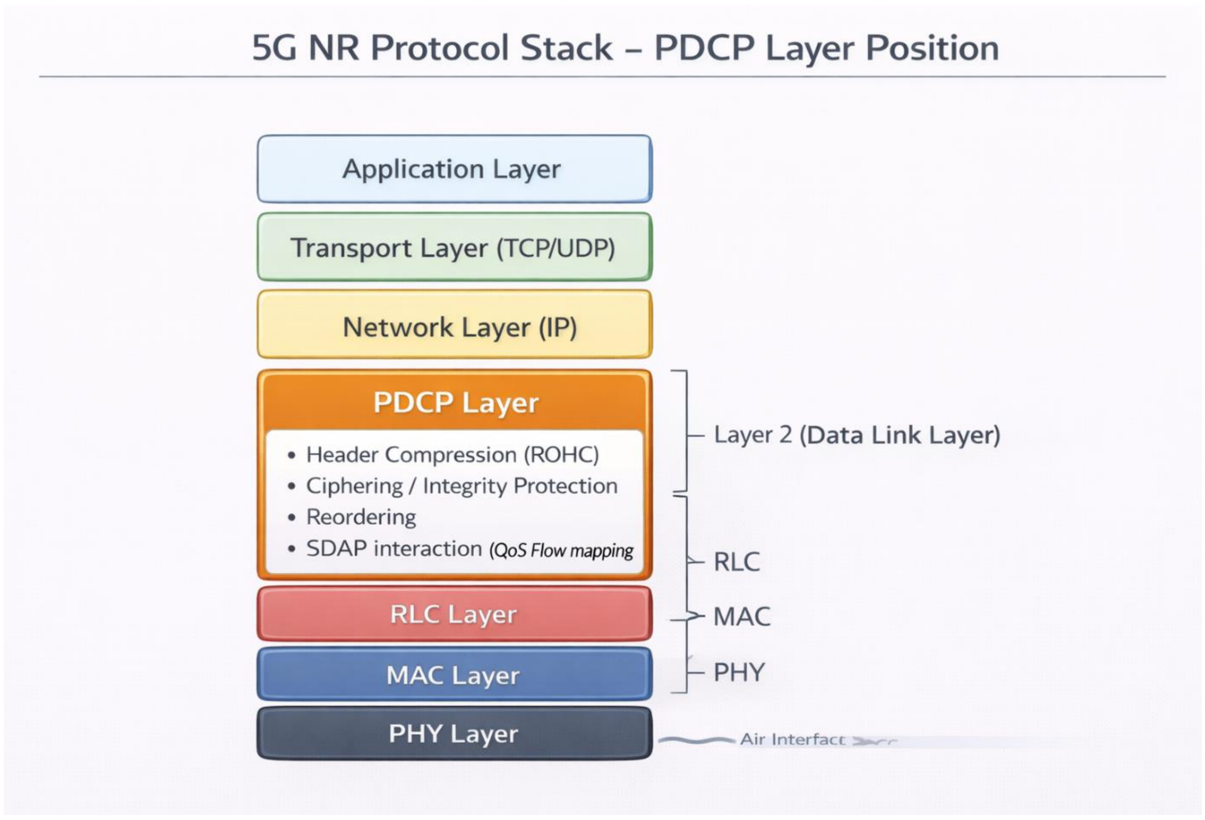
2. 5G NR PROTOCOL STACK OVERVIEW

The 5G NR protocol stack defines how data is processed and transmitted between the User Equipment (UE) and the gNB. Each layer in the stack performs specific functions to ensure reliable and efficient communication over the wireless network.

The protocol stack consists of higher layers and lower layers. The higher layers handle application and control signaling, while the lower layers manage data transmission over the air interface. The PDCP layer is positioned between the higher layers and the RLC layer, acting as an interface that provides security and data optimization before transmission.

The main layers involved in the 5G NR protocol stack are listed below:

LAYER	FUNCTION
PDCP	Header compression, ciphering, integrity protection
RLC	Segmentation, reassembly, retransmission
MAC	Scheduling and resource allocation
PHY	Physical transmission over radio channel


5G NR Protocol Stack – PDCP Layer Position
Application Layer
Transport Layer (TCP/UDP)
Network Layer (IP)
PDCP Layer
• Header Compression (ROHC)
• Ciphering / Integrity Protection
• Reordering
• SDAP interaction (QoS Flow mapping
RLC Layer
MAC Layer
PHY Layer
Layer 2 (Data Link Layer)
RLC
MAC
PHY
Air Interface

3. PDCP LAYER FUNCTIONS AND RESPONSIBILITIES

The Packet Data Convergence Protocol (PDCP) layer plays an important role in ensuring efficient, secure, and reliable data transmission in the 5G NR protocol stack. It receives data packets from higher layers and prepares them for transmission by performing processing functions before forwarding them to the RLC layer.

One of the primary responsibilities of the PDCP layer is header compression, which reduces unnecessary header information in IP packets and improves bandwidth utilization. The PDCP layer also performs ciphering to protect user data confidentiality and integrity protection to ensure that control messages are not altered during transmission.

In addition, the PDCP layer assigns sequence numbers to packets for proper tracking and performs duplicate detection and packet reordering at the receiver side. This ensures that packets are delivered in the correct order even when transmission delays occur. By combining these functions, the PDCP layer improves transmission efficiency while maintaining data security and reliability in 5G communication systems.

4. SECURITY OPERATIONS

Security is one of the major responsibilities of the PDCP layer in 5G NR communication. The PDCP layer ensures that transmitted data is protected from unauthorized access and modification during transmission between the UE and the network.

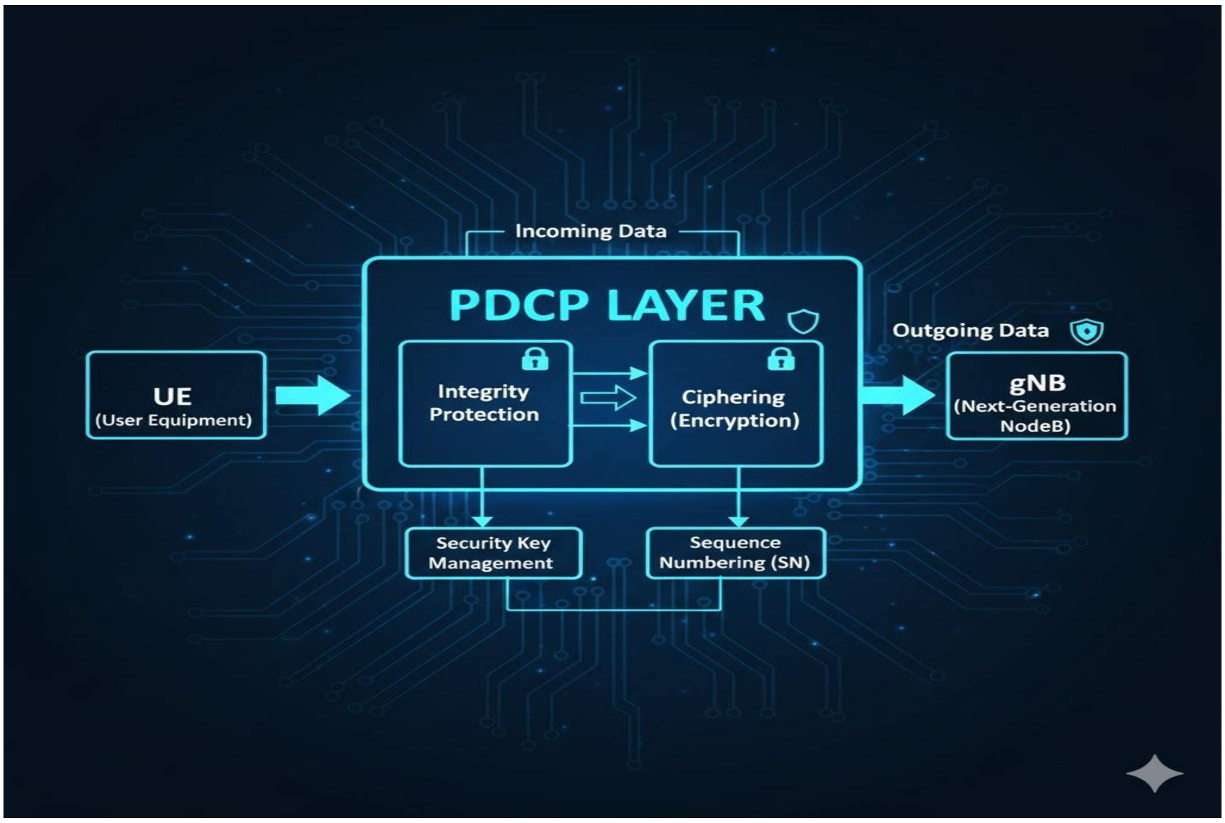
4.1 Ciphering

Ciphering is used to maintain data confidentiality by encrypting user data and signaling messages before transmission. The PDCP layer applies encryption algorithms using security keys provided during the authentication process. At the receiver side, the data is decrypted using the corresponding key to recover the original information.

4.2 Integrity Protection

Integrity protection is mainly applied to control plane messages to ensure that the transmitted data has not been modified or tampered with during transmission. The receiver verifies the integrity of the message using an integrity check value generated at the transmitter side.

These security mechanisms help maintain secure communication in the 5G network while preventing data leakage and unauthorized modifications.



5. HEADER COMPRESSION USING RoHC

In wireless communication systems, IP packets contain large headers compared to the actual payload, especially in applications such as Voice over IP (VoIP) and real-time video streaming. Transmitting full headers over the radio interface increases bandwidth usage and reduces transmission efficiency. To overcome this issue, the PDCP layer uses Robust Header Compression (RoHC) to reduce header size before transmission.

RoHC works by identifying fields in the packet header that remain constant or change predictably during a session. Instead of transmitting the complete header every time, only the changing information is sent, while the remaining header fields are reconstructed at the receiver using stored context information. This significantly reduces the number of bits transmitted over the air interface and improves overall spectral efficiency.

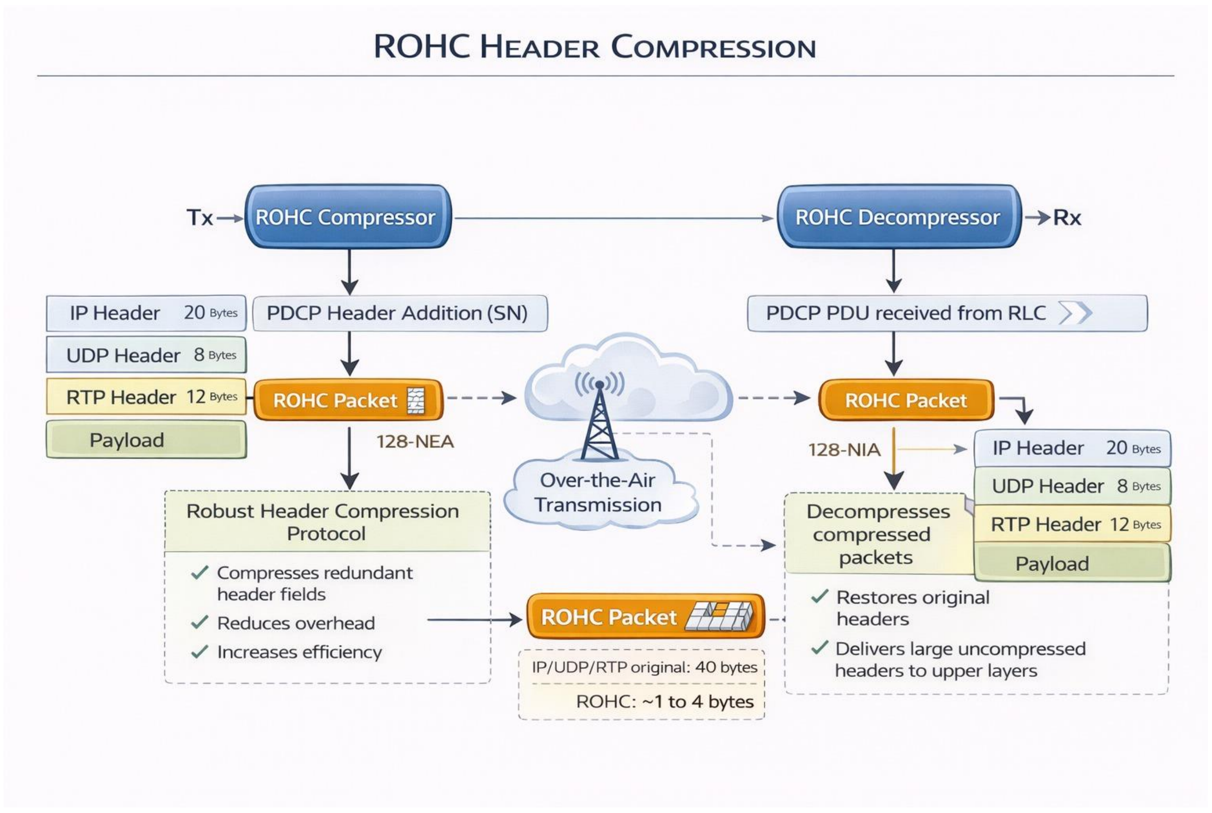
5.1 Working Principle of RoHC

During the initial stage, the full header is transmitted to establish a context between the transmitter and receiver. Once the context is created, only compressed headers are transmitted. The receiver uses the stored context to rebuild the original header accurately. If synchronization is lost, the system automatically refreshes the context by sending a larger header again.

5.2 Advantages of Header Compression

- Reduces packet size and transmission overhead
- Improves bandwidth utilization
- Supports efficient real-time communication
- Reduces transmission delay in wireless networks

The implementation of RoHC in the PDCP layer plays an important role in maintaining efficient data transmission while supporting high data rate services in 5G networks.



6. PDCP REORDERING AND DUPLICATE DETECTION

During wireless transmission, packets may reach the receiver in a different order due to retransmissions, varying channel conditions, or scheduling delays in lower layers. The PDCP layer ensures that packets are delivered to higher layers in the correct order by performing reordering and duplicate detection functions.

Each packet processed by the PDCP layer is assigned a sequence number at the transmitter. At the receiver side, these sequence numbers are checked to identify missing or out-of-order packets. Packets that arrive earlier than expected are temporarily stored in a reordering buffer until the correct sequence is restored. This helps maintain proper data flow for applications that require ordered delivery.

6.1 Reordering Mechanism

The PDCP layer maintains a reordering window where incoming packets are arranged based on their sequence numbers. If a packet is delayed, the receiver waits for a certain duration before forwarding the available packets to higher layers. This prevents incorrect packet delivery and improves communication reliability.

6.2 Duplicate Detection

Duplicate packets may occur due to retransmissions at lower layers such as RLC or MAC. The PDCP layer checks sequence numbers to identify repeated packets and discards them. This avoids unnecessary processing and ensures efficient utilization of network resources.

By performing reordering and duplicate detection, the PDCP layer maintains data consistency, reduces errors, and ensures smooth packet delivery in 5G communication systems.

7. SRB VS DRB PROCESSING

In the 5G NR protocol stack, the PDCP layer processes two types of radio bearers known as Signaling Radio Bearers (SRB) and Data Radio Bearers (DRB). These bearers are used to separate control signaling information from user data, allowing the network to apply appropriate security and processing mechanisms for different types of communication.

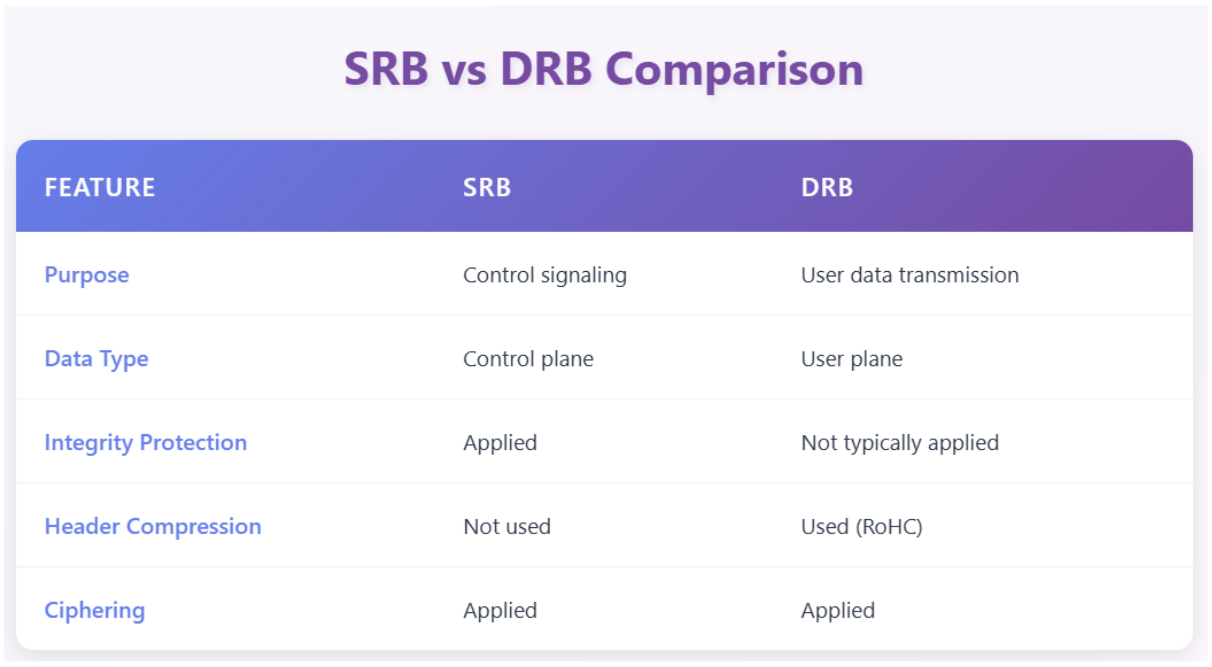
7.1 Signaling Radio Bearer (SRB)

Signaling Radio Bearers are used for transmitting control plane messages between the User Equipment (UE) and the network. These messages include connection establishment, authentication, configuration updates, and mobility-related signaling. Since signaling information is critical for network operation, the PDCP layer applies both ciphering and integrity protection to ensure that the messages are secure and not modified during transmission.

7.2 Data Radio Bearer (DRB)

Data Radio Bearers are used for transmitting user plane data such as internet traffic, voice, and video communication. In DRB processing, the PDCP layer performs header compression to reduce packet size and improve bandwidth efficiency. Ciphering is also applied to maintain confidentiality of user data, while integrity protection is generally not required for user plane traffic.

7.3 Difference Between SRB and DRB



SRB vs DRB Comparison

FEATURE	SRB	DRB
Purpose	Control signaling	User data transmission
Data Type	Control plane	User plane
Integrity Protection	Applied	Not typically applied
Header Compression	Not used	Used (RoHC)
Ciphering	Applied	Applied

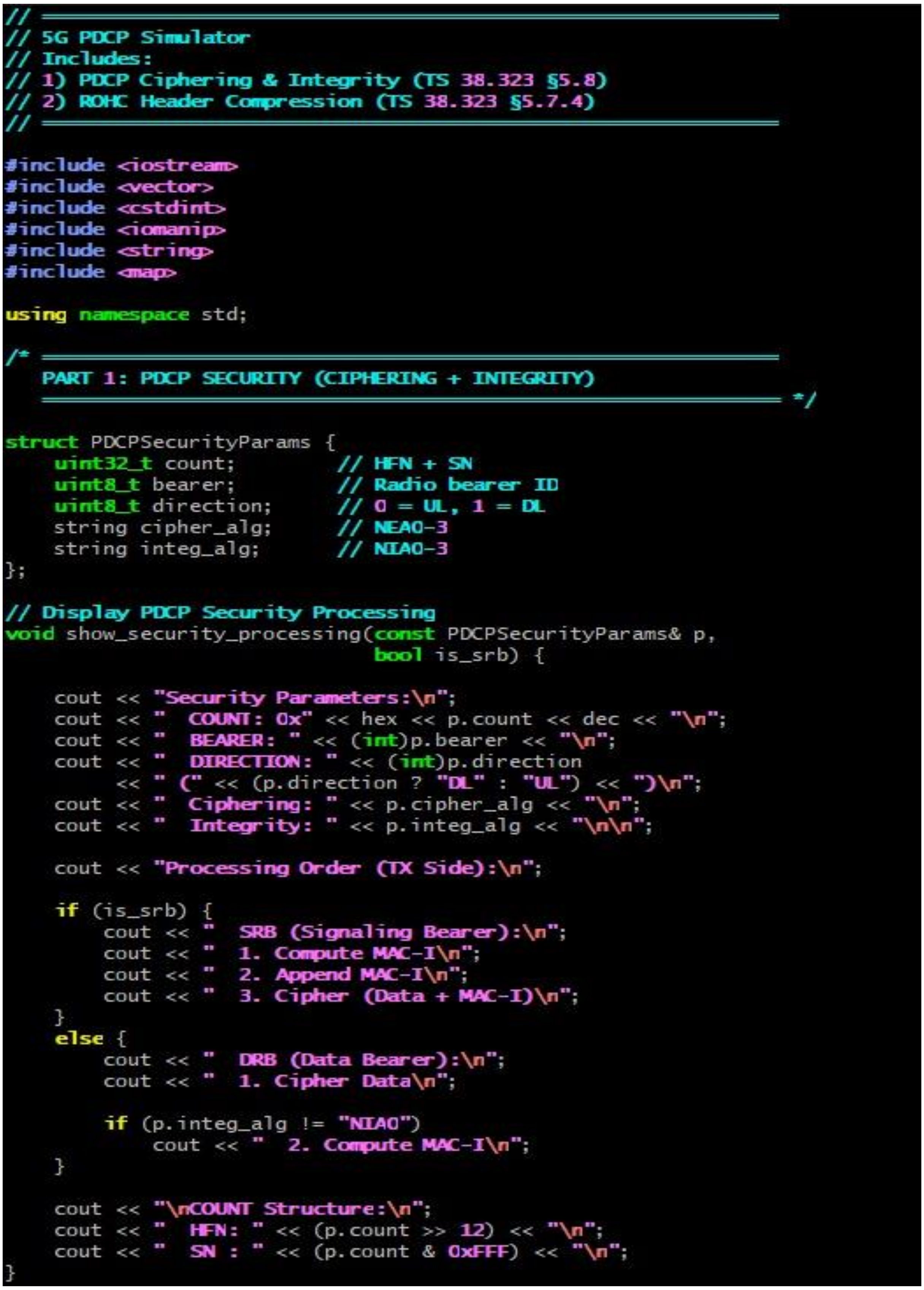
8. C++ IMPLEMENTATION

The PDCP layer functionalities described in this project are implemented using a C++ based simulation program. The implementation demonstrates key PDCP operations such as header compression, security processing, packet sequencing, reordering, and duplicate detection. The simulator models both SRB and DRB processing to represent control plane and user plane behavior in a simplified 5G environment. The objective of this implementation is to provide a practical understanding of PDCP internal processing through step-by-step execution.

8.1 Step 1 : PDCP Security (Ciphering + Integrity)

This part of the code models the PDCP security configuration used for ciphering and integrity protection in 5G NR. The PDCP Security Params structure encapsulates all mandatory security inputs: COUNT, bearer identity, transmission direction, and the selected ciphering and integrity algorithms. The COUNT value, formed as HFN concatenated with SN, ensures replay protection and keystream uniqueness, while the bearer and direction fields guarantee separation of security contexts across radio bearers and between uplink and downlink transmissions. These parameters collectively act as inputs to the PDCP security algorithms defined in the specification.

The processing logic reflects the correct transmission-side order defined in the standard. For SRBs (control plane), integrity protection is applied first by computing and appending the MAC-I, followed by ciphering of the data and integrity field together. For DRBs (user plane), only ciphering is performed, as integrity protection is not applied under normal NR operation. By explicitly showing the COUNT breakdown into HFN and SN and distinguishing SRB and DRB procedures, this code section accurately represents the PDCP security workflow defined in TS 38.323.

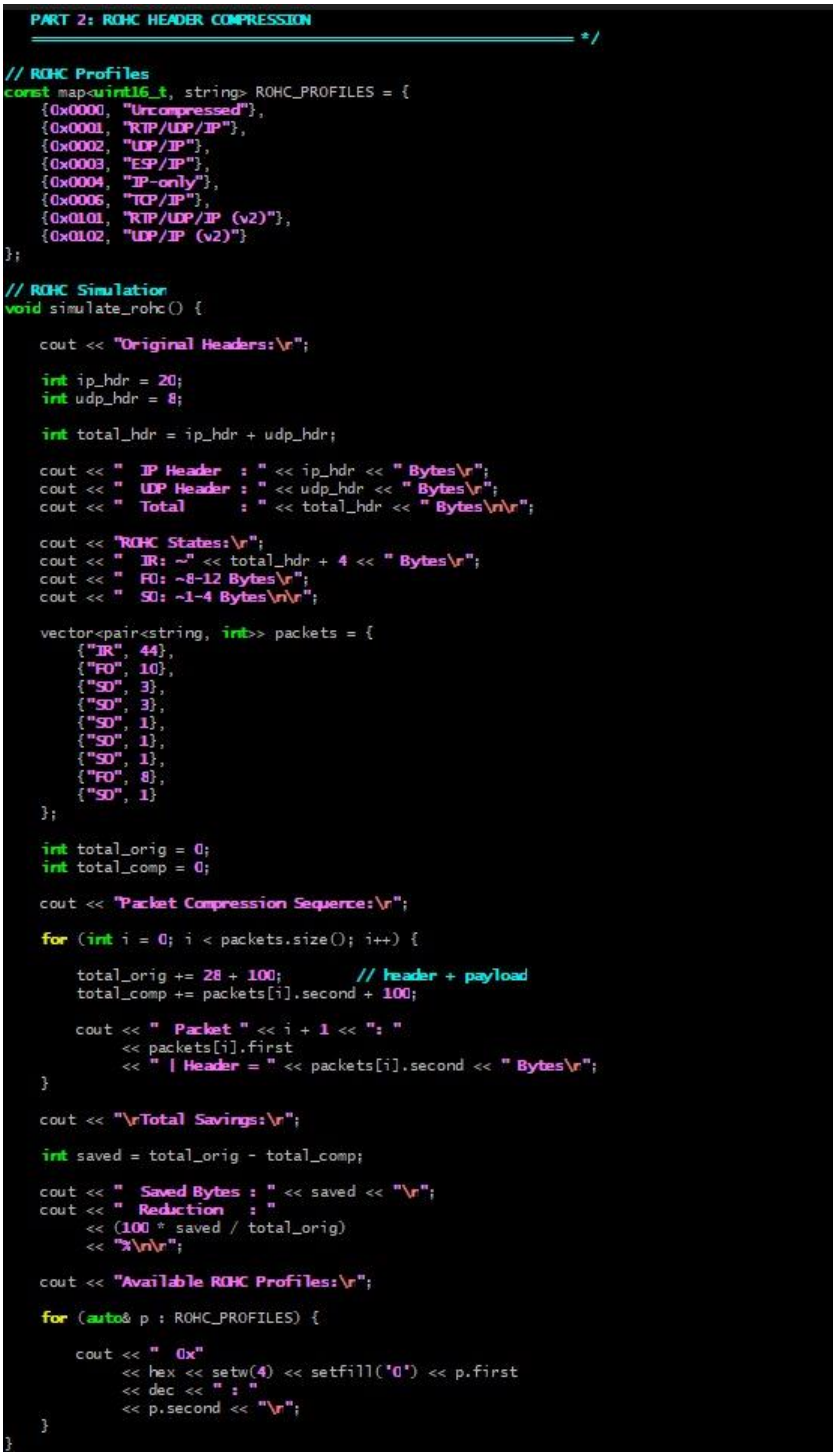


```
// ════════════════════════════════════════════════════════════════
// 5G PDCP Simulator
// Includes:
// 1) PDCP Ciphering & Integrity (TS 38.323 §5.8)
// 2) ROHC Header Compression (TS 38.323 §5.7.4)
// ════════════════════════════════════════════════════════════════

#include <iostream>
#include <vector>
#include <cstdint>
#include <iomanip>
#include <string>
#include <map>

using namespace std;

/* ════════════════════════════════════════════════════════════════
   PART 1: PDCP SECURITY (CIPHERING + INTEGRITY)
   ════════════════════════════════════════════════════════════════ */

struct PDCPSecurityParams {
    uint32_t count;          // HFN + SN
    uint8_t bearer;          // Radio bearer ID
    uint8_t direction;       // 0 = UL, 1 = DL
    string cipher_alg;       // NEA0-3
    string integ_alg;        // NIA0-3
};

// Display PDCP Security Processing
void show_security_processing(const PDCPSecurityParams& p,
                              bool is_srb) {

    cout << "Security Parameters:\n";
    cout << "  COUNT: 0x" << hex << p.count << dec << "\n";
    cout << "  BEARER: " << (int)p.bearer << "\n";
    cout << "  DIRECTION: " << (int)p.direction
         << " (" << (p.direction ? "DL" : "UL") << ")\n";
    cout << "  Ciphering: " << p.cipher_alg << "\n";
    cout << "  Integrity: " << p.integ_alg << "\n\n";

    cout << "Processing Order (TX Side):\n";

    if (is_srb) {
        cout << "  SRB (Signaling Bearer):\n";
        cout << "  1. Compute MAC-I\n";
        cout << "  2. Append MAC-I\n";
        cout << "  3. Cipher (Data + MAC-I)\n";
    }
    else {
        cout << "  DRB (Data Bearer):\n";
        cout << "  1. Cipher Data\n";

        if (p.integ_alg != "NIA0")
            cout << "  2. Compute MAC-I\n";
    }

    cout << "\nCOUNT Structure:\n";
    cout << "  HFN: " << (p.count >> 12) << "\n";
    cout << "  SN : " << (p.count & 0xFFF) << "\n";
}
```

8.2 Step 2 : ROHC Header Compression

This part of the code implements a simplified simulation of ROHC (Robust Header Compression) as performed by the PDCP layer in 5G NR. ROHC is used to reduce the size of repetitive IP/UDP/RTP headers in packet-based communication, thereby improving spectral efficiency over the air interface. The code defines a set of ROHC profiles mapped to profile identifiers, representing different protocol combinations such as RTP/UDP/IP and UDP/IP. These profiles reflect the standardized ROHC profile types supported in NR, where the compressor and decompressor agree on a specific profile depending on the upper-layer protocol stack. By listing and displaying these profiles, the implementation demonstrates how PDCP selects and applies appropriate header compression schemes.

The simulation function models the transition from uncompressed headers to compressed ROHC states, such as IR (Initialization and Refresh), FO (First Order), and SO (Second Order). Initially, the full IP and UDP headers are considered, and the total header size is calculated. As packets are processed sequentially, the header size reduces significantly depending on the compression state, illustrating how ROHC minimizes redundant information after context establishment. The code then computes total original size versus compressed size and calculates the overall byte savings and reduction percentage. This reflects the practical benefit of ROHC in NR systems, where repeated header fields are not retransmitted in full for every packet, resulting in substantial bandwidth savings while maintaining reliable decompression at the receiver.

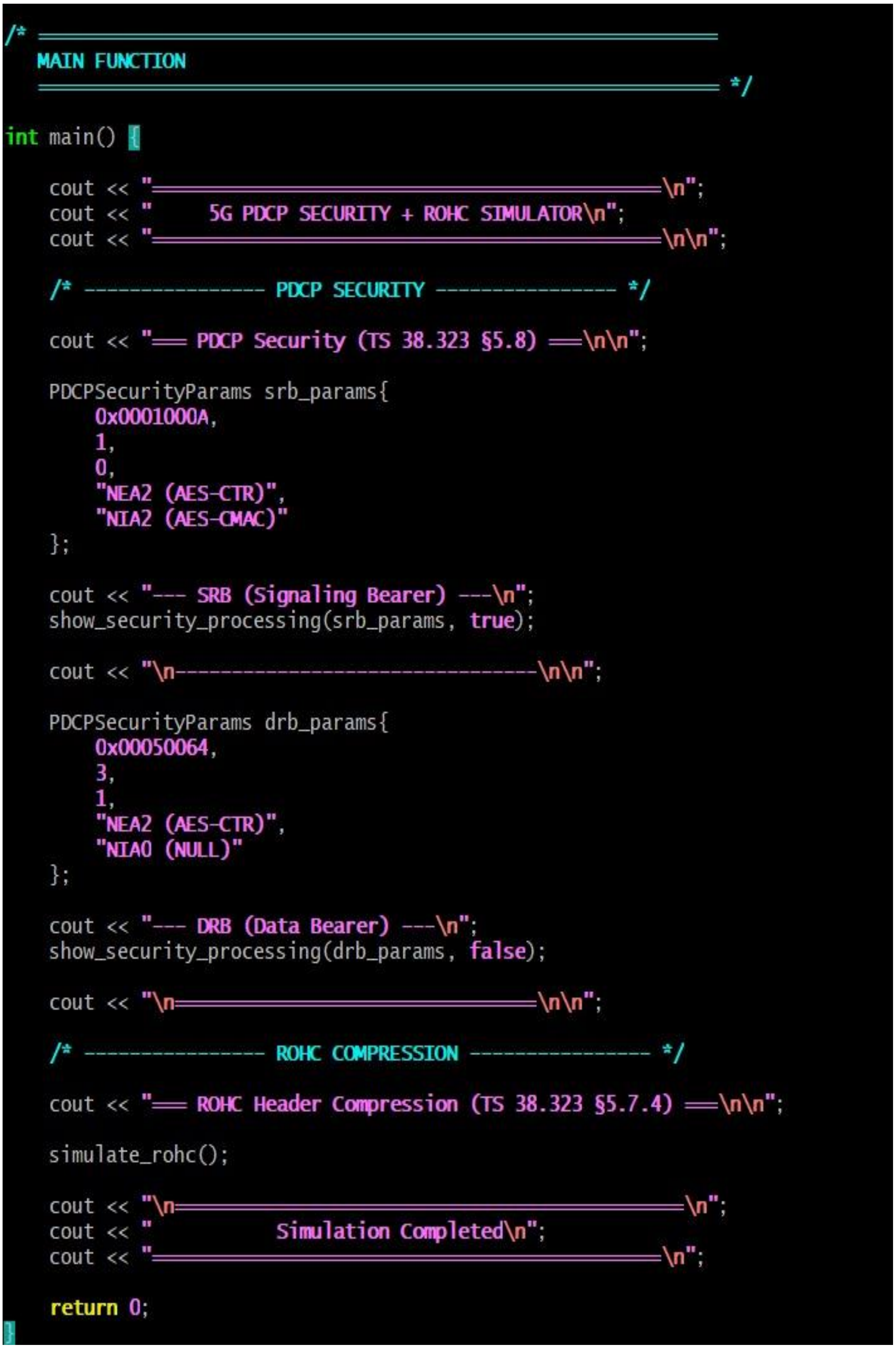


```
   PART 2: ROHC HEADER COMPRESSION
   ============================================================ */

// ROHC Profiles
const map<uint16_t, string> ROHC_PROFILES = {
    {0x0000, "Uncompressed"},
    {0x0001, "RTP/UDP/IP"},
    {0x0002, "UDP/IP"},
    {0x0003, "ESP/IP"},
    {0x0004, "IP-only"},
    {0x0006, "TCP/IP"},
    {0x0101, "RTP/UDP/IP (v2)"},
    {0x0102, "UDP/IP (v2)"}
};

// ROHC Simulation
void simulate_rohc() {

    cout << "Original Headers:\r";

    int ip_hdr = 20;
    int udp_hdr = 8;

    int total_hdr = ip_hdr + udp_hdr;

    cout << "  IP Header  : " << ip_hdr << " Bytes\r";
    cout << "  UDP Header : " << udp_hdr << " Bytes\r";
    cout << "  Total      : " << total_hdr << " Bytes\n\r";

    cout << "ROHC States:\r";
    cout << "  IR: ~" << total_hdr + 4 << " Bytes\r";
    cout << "  FO: ~8-12 Bytes\r";
    cout << "  SO: ~1-4 Bytes\n\r";

    vector<pair<string, int>> packets = {
        {"IR", 44},
        {"FO", 10},
        {"SO", 3},
        {"SO", 3},
        {"SO", 1},
        {"SO", 1},
        {"SO", 1},
        {"FO", 8},
        {"SO", 1}
    };

    int total_orig = 0;
    int total_comp = 0;

    cout << "Packet Compression Sequence:\r";

    for (int i = 0; i < packets.size(); i++) {

        total_orig += 28 + 100;          // header + payload
        total_comp += packets[i].second + 100;

        cout << "  Packet " << i + 1 << ": "
             << packets[i].first
             << " | Header = " << packets[i].second << " Bytes\r";
    }

    cout << "\rTotal Savings:\r";

    int saved = total_orig - total_comp;

    cout << "  Saved Bytes : " << saved << "\r";
    cout << "  Reduction   : "
       << (100 * saved / total_orig)
       << "%\n\r";

    cout << "Available ROHC Profiles:\r";

    for (auto& p : ROHC_PROFILES) {

        cout << "  0x"
             << hex << setw(4) << setfill('0') << p.first
             << dec << " : "
             << p.second << "\r";
    }
}
```

8.3 Step 3 : Main Function

This main function integrates and demonstrates the complete PDCP security and ROHC processing workflow in a structured simulation format. It initializes two separate PDCP security contexts: one for an SRB (Signalling Radio Bearer) and another for a DRB (Data Radio Bearer). For the SRB case, integrity protection and ciphering are both configured, reflecting control-plane security requirements where messages must be authenticated and encrypted. For the DRB case, only ciphering is enabled while integrity is set to NULL, aligning with NR user-plane behavior where confidentiality is mandatory but integrity protection is generally not applied. By invoking the security processing function for both bearer types, the main routine clearly demonstrates the difference in processing order and algorithm usage as defined in the specification.

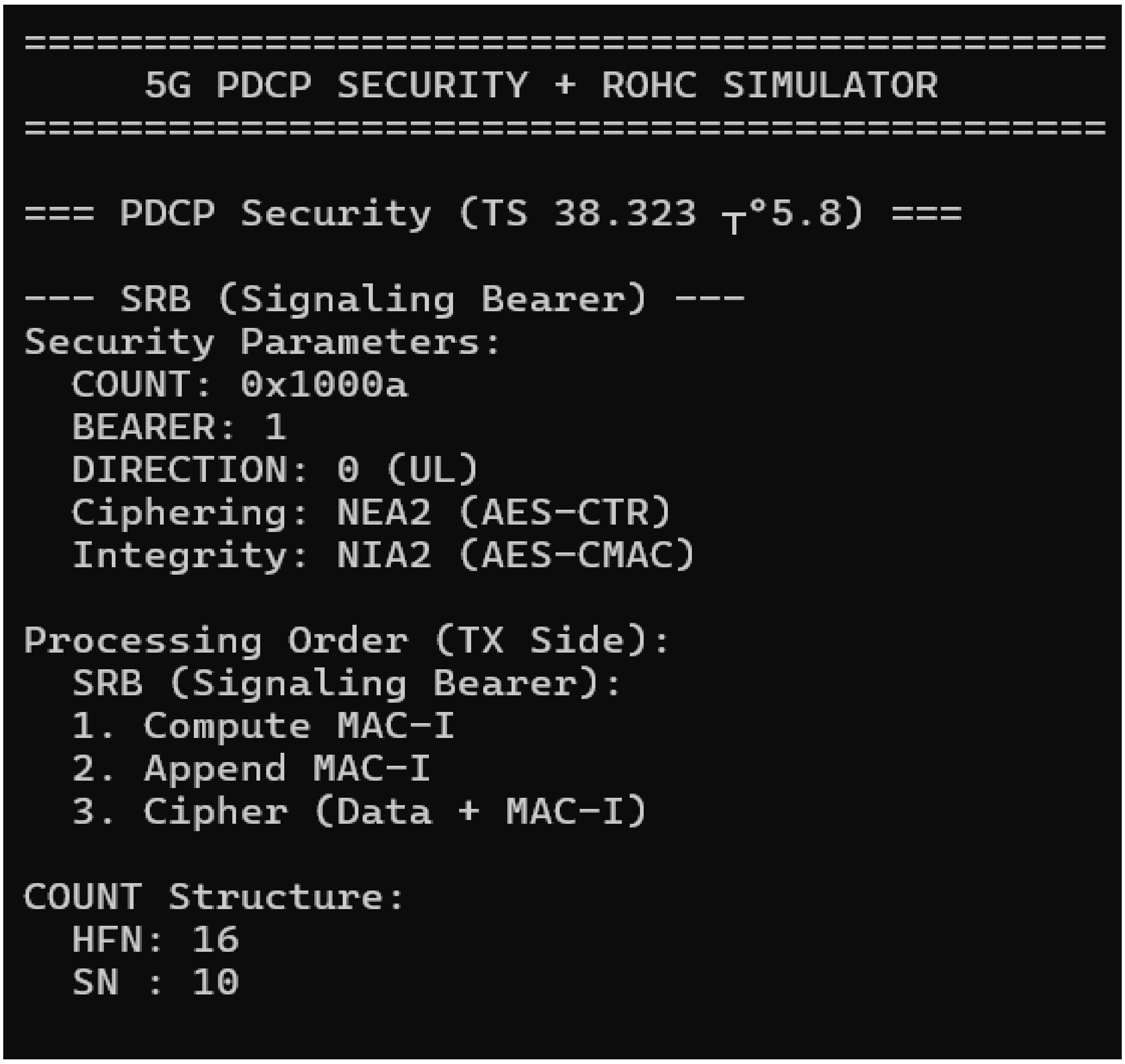
Following the security demonstration, the function transitions to ROHC header compression simulation as defined under the PDCP header compression procedures in the same specification. It invokes the ROHC simulation function to illustrate header size reduction and compression state transitions. Structurally, this main function serves as the execution driver of the simulator, sequentially presenting PDCP security (ciphering and integrity) followed by header compression, thereby modeling the two major functional responsibilities of the NR PDCP layer. The organized output formatting also mirrors a practical test or validation flow, helping to conceptually link specification-defined procedures with their simulated implementation.



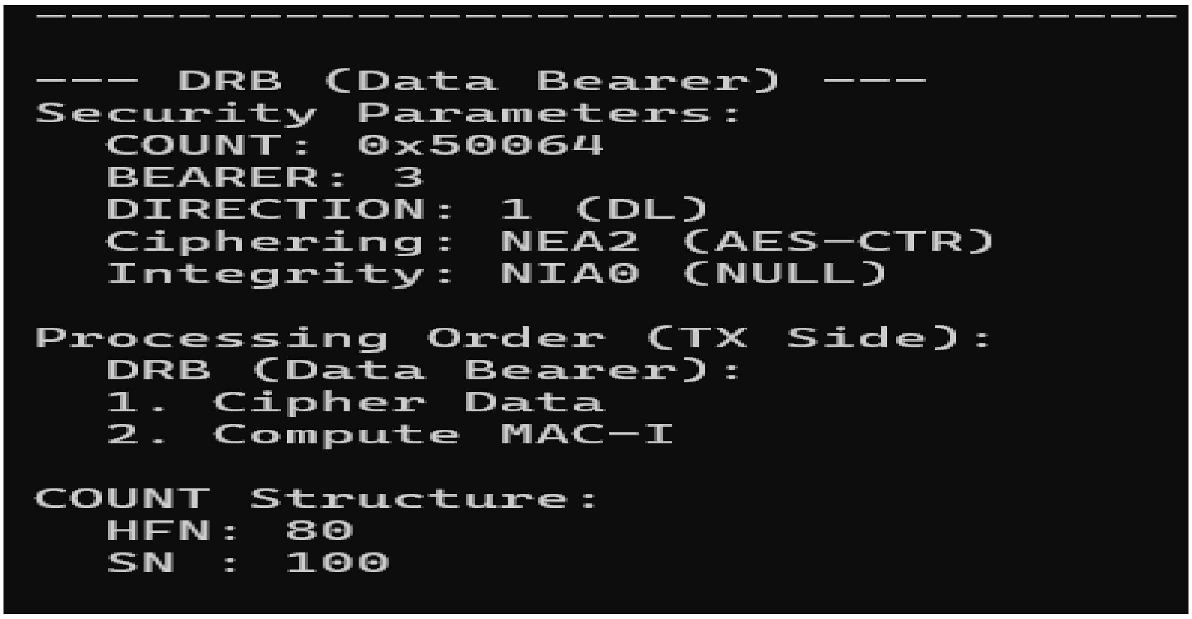
```cpp
/* ═══════════════════════════════════════════════════════════
   MAIN FUNCTION
   ═══════════════════════════════════════════════════════════ */

int main() {

    cout << "════════════════════════════════════════════\n";
    cout << "     5G PDCP SECURITY + ROHC SIMULATOR\n";
    cout << "════════════════════════════════════════════\n\n";

    /* ---------------- PDCP SECURITY ---------------- */

    cout << "══ PDCP Security (TS 38.323 §5.8) ══\n\n";

    PDCPSecurityParams srb_params{
        0x0001000A,
        1,
        0,
        "NEA2 (AES-CTR)",
        "NIA2 (AES-CMAC)"
    };

    cout << "--- SRB (Signaling Bearer) ---\n";
    show_security_processing(srb_params, true);

    cout << "\n--------------------------------\n\n";

    PDCPSecurityParams drb_params{
        0x00050064,
        3,
        1,
        "NEA2 (AES-CTR)",
        "NIA0 (NULL)"
    };

    cout << "--- DRB (Data Bearer) ---\n";
    show_security_processing(drb_params, false);

    cout << "\n════════════════════════════════\n\n";

    /* ---------------- ROHC COMPRESSION ---------------- */

    cout << "══ ROHC Header Compression (TS 38.323 §5.7.4) ══\n\n";

    simulate_rohc();

    cout << "\n════════════════════════════════════════════\n";
    cout << "           Simulation Completed\n";
    cout << "════════════════════════════════════════════\n";

    return 0;
}
```

9. SAMPLE OUTPUT

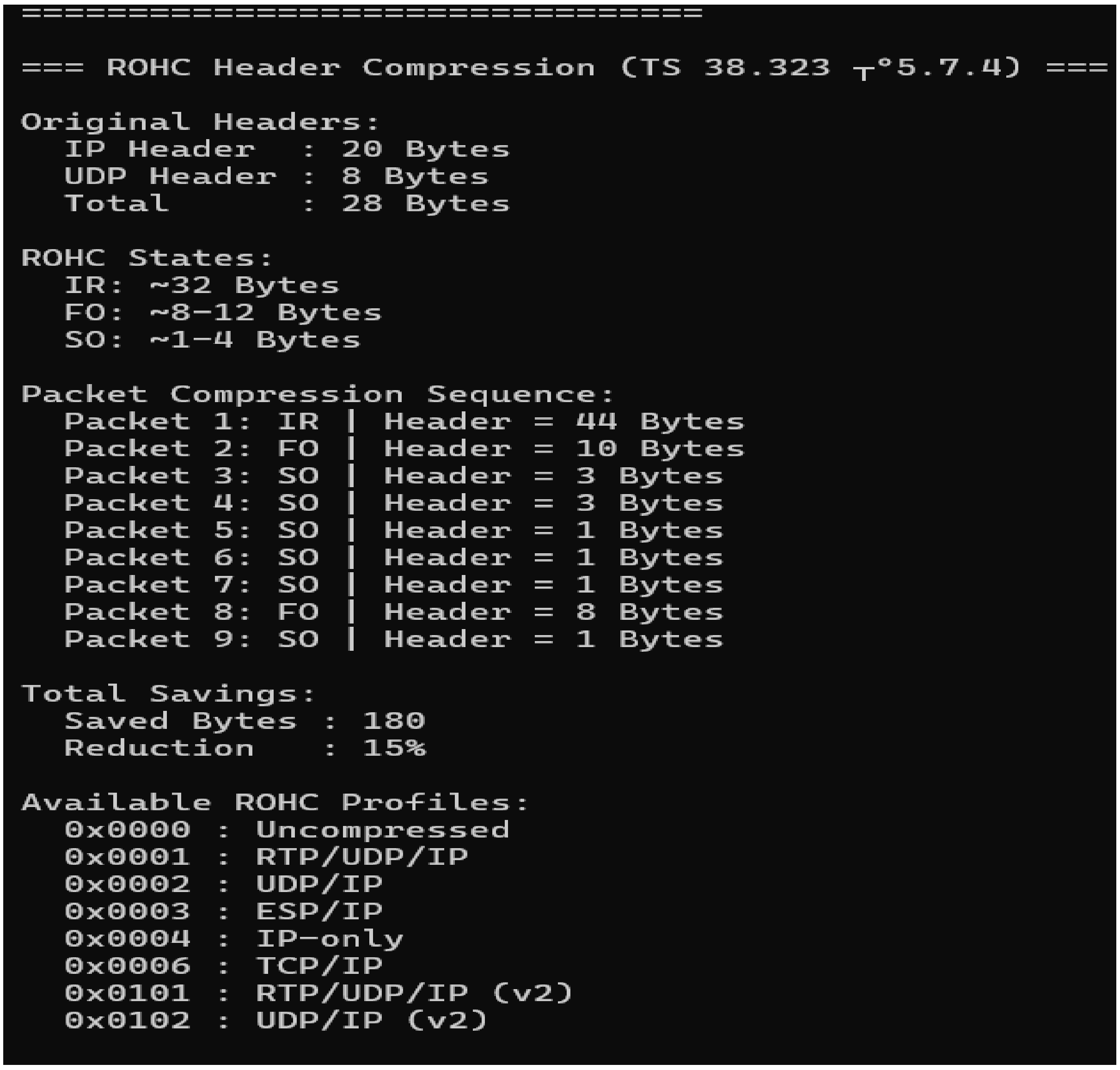
This section shows the output generated from the PDCP simulator during execution. The output demonstrates the application of ciphering, integrity protection, and header compression for SRB and DRB transmissions. It also shows packet reordering and duplicate detection at the receiver side. The execution logs help in understanding the step-by-step processing of PDCP operations. The final statistics confirm correct packet transmission and reception in the simulated environment.



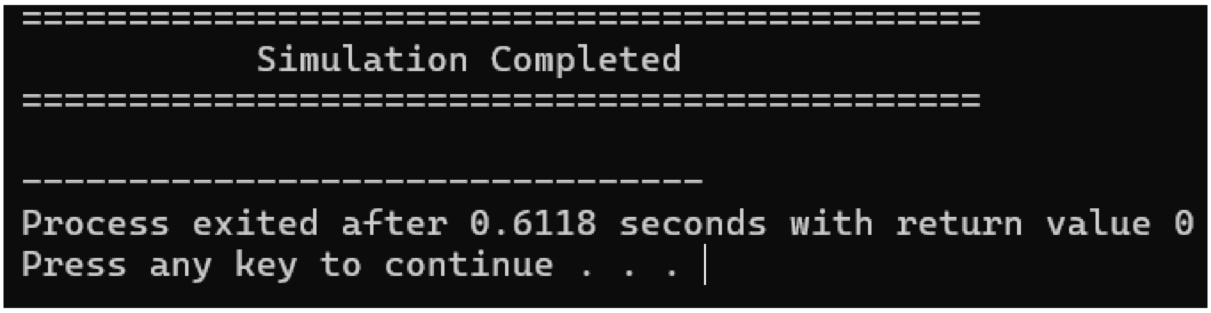
```
=============================================
     5G PDCP SECURITY + ROHC SIMULATOR
=============================================

=== PDCP Security (TS 38.323 ┬°5.8) ===

--- SRB (Signaling Bearer) ---
Security Parameters:
  COUNT: 0x1000a
  BEARER: 1
  DIRECTION: 0 (UL)
  Ciphering: NEA2 (AES-CTR)
  Integrity: NIA2 (AES-CMAC)

Processing Order (TX Side):
  SRB (Signaling Bearer):
  1. Compute MAC-I
  2. Append MAC-I
  3. Cipher (Data + MAC-I)

COUNT Structure:
  HFN: 16
  SN : 10
```



```
 ----------------------------------------

 --- DRB (Data Bearer) ---
 Security Parameters:
    COUNT: 0x50064
    BEARER: 3
    DIRECTION: 1 (DL)
    Ciphering: NEA2 (AES-CTR)
    Integrity: NIA0 (NULL)

 Processing Order (TX Side):
    DRB (Data Bearer):
    1. Cipher Data
    2. Compute MAC-I

 COUNT Structure:
    HFN: 80
    SN : 100
```



```
================================

=== ROHC Header Compression (TS 38.323 ┬°5.7.4) ===

Original Headers:
  IP Header  : 20 Bytes
  UDP Header : 8 Bytes
  Total      : 28 Bytes

ROHC States:
  IR: ~32 Bytes
  FO: ~8-12 Bytes
  SO: ~1-4 Bytes

Packet Compression Sequence:
  Packet 1: IR | Header = 44 Bytes
  Packet 2: FO | Header = 10 Bytes
  Packet 3: SO | Header = 3 Bytes
  Packet 4: SO | Header = 3 Bytes
  Packet 5: SO | Header = 1 Bytes
  Packet 6: SO | Header = 1 Bytes
  Packet 7: SO | Header = 1 Bytes
  Packet 8: FO | Header = 8 Bytes
  Packet 9: SO | Header = 1 Bytes

Total Savings:
  Saved Bytes : 180
  Reduction   : 15%

Available ROHC Profiles:
  0x0000 : Uncompressed
  0x0001 : RTP/UDP/IP
  0x0002 : UDP/IP
  0x0003 : ESP/IP
  0x0004 : IP-only
  0x0006 : TCP/IP
  0x0101 : RTP/UDP/IP (v2)
  0x0102 : UDP/IP (v2)
```



```
=============================================
           Simulation Completed
=============================================

--------------------------------
Process exited after 0.6118 seconds with return value 0
Press any key to continue . . .
```

10. KEYCONCEPTS

The project is based on important PDCP layer concepts used in 5G NR communication. These include header compression to reduce transmission overhead, ciphering for data confidentiality, integrity protection for secure signaling, sequence numbering for packet tracking, and reordering with duplicate detection to ensure reliable data delivery. These concepts help improve efficiency, security, and reliability in wireless communication.

11. COMPILATION & EXECUTION GUIDE

11.1 Prerequisites :

This simulation requires a C++11 (or later) compatible compiler. Recommended compilers include GCC 7+, Clang 5+, or MSVC 2017+. The program uses only the C++ Standard Library, so no external libraries are required.

11.2 Compilation Commands

- Windows (MinGW / Git Bash):

```
g++ code_final.cpp -o Test4_PDCP
./Test4_PDCP.exe
```

- Linux / macOS:

```
g++ code_final.cpp -o Test4_PDCP
./Test4_PDCP
```

- Windows (MSVC):

```
cl /EHsc code_final.cpp
/Fe:Test4_PDCP.exe
```

12. CONCLUSION :

The PDCP layer simulator developed in this project demonstrates the essential functionalities of the PDCP layer in a simplified 5G NR environment. The implementation successfully illustrates important PDCP operations such as header compression, ciphering, integrity protection, sequence numbering, packet reordering, and duplicate detection. These functions help improve transmission efficiency while ensuring secure and reliable communication between the UE and the network.

Through the C++ based simulation, the project provides practical insight into how PDCP processes data before transmission and how packets are handled at the receiver side. The results obtained from the simulation confirm correct processing of both SRB and DRB traffic scenarios. Overall, the project helps in understanding the role of the PDCP layer in enhancing performance, security, and reliability in modern 5G communication systems.