ASN.1 Encoder/Decoder API Documentation For SIB1 Transmission Over TCP

1. Overview

This document provides a structured and submission-ready reference for the APIs used to encode, transmit, and decode a 5G NR SIB1 message using ASN.1 (via asn1c) over a TCP client-server model. It includes socket APIs, ASN.1 encoding/decoding APIs, key data structures, and the full message handling flow.

2. Socket APIs (POSIX)

API	Description
socket()	Create a TCP socket.
bind()	Bind socket to an IP address and port (server).
listen()	Enable passive listening mode (server).
accept()	Accept incoming client connections.
connect()	Connect to remote server (client).
read()	Receive data from a socket.
write()	Send data over TCP.
close()	Close a file descriptor or socket.
inet_pton()	Convert IP string (text) to binary form.
inet_ntop()	Convert binary IP to readable string.

3. ASN.1 Encoding & Decoding APIs (asn1c)

API	Purpose
uper_encode_to_buffer()	Encode a PDU using Unaligned PER (UPER).
uper_decode()	Decode UPER bytes into ASN.1 structure.
xer_fprint()	Print decoded ASN.1 structure in human-readable XML format.
ASN_SEQUENCE_ADD()	Append an element into ASN.1 SEQUENCE OF / SET OF.
ASN_STRUCT_FREE()	Free memory allocated for PDU structures.

4. Key ASN.1 Data Structures (Generated by asn1c)

These data structures are automatically generated by asn1c from the NR-RRC ASN.1 specification (3GPP TS 38.331).
They represent the decoded in-memory format of the SIB1 message:
• BCCH_DL_SCH_Message_t
• SIB1_t
• CellAccessRelatedInfo_t
• PLMN_IdentityInfoList_t
• PLMN_IdentityInfo_t
• PLMN_Identity_t
• MCC_t, MNC_t

5. Encoder and Decoder Processing Flow

The following outlines the full pipeline used for constructing and transmitting SIB1 across systems:

Encoder (System A):
1. Create BCCH_DL_SCH_Message PDU structure.
2. Populate SIB1 fields (q-RxLevMin, PLMN, MCC, MNC, CellIdentity, etc.).
3. Print intermediate XML using xer_fprint() for debugging.
4. Encode using uper_encode_to_buffer().
5. Open TCP connection and send:
• 4-byte length header
• Encoded PER payload
6. Close connection.

Decoder (System B):
1. Accept incoming TCP connection.
2. Read 4-byte length header.
3. Receive raw PER payload bytes.
4. Decode using uper_decode().
5. Dump XML using xer_fprint().
6. Free all allocated structures.

6. Return Structures

asn_enc_rval_t (encoding result):
• encoded → number of bits produced, or -1 on failure

asn_dec_rval_t (decoding result):
• code → RC_OK, RC_FAIL, or RC_WMORE
• consumed → number of bits consumed

7. Summary

This document provides all the required APIs and descriptions used to implement a full SIB1 encoder/decoder pipeline over TCP using ASN.1 PER encoding. It is suitable for academic submissions, reports, or technical documentation.