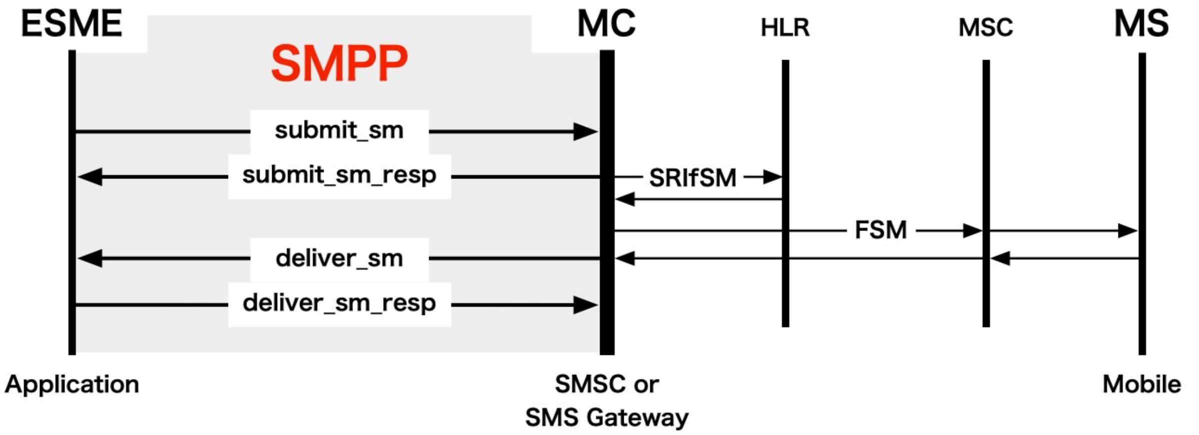
How SMPP Works to deliver SMS

This diagram explains how **SMPP** (Short Message Peer-to-Peer Protocol) is used for communication between an **ESME** (External Short Message Entity), the **MC** (Message Center or SMS Gateway), and components like **HLR** (Home Location Register), **MSC** (Mobile Switching Center), and the **MS** (Mobile Station).



Here’s a detailed breakdown:

Entities in the Diagram

1. **ESME (External Short Message Entity)**:
    - This represents an application or system that connects to the SMSC (Short Message Service Center) to send and receive SMS messages.
    - Examples: Bulk SMS applications, notifications systems, chatbots, etc.
2. **MC (Message Center or SMS Gateway)**:
    - The SMSC or SMS Gateway acts as the intermediary between the ESME and the mobile network.
    - It handles incoming and outgoing messages, delivery receipts (DLRs), and ensures message delivery.
3. **HLR (Home Location Register)**:
    - A database in a mobile network that holds details of subscriber information, including location and routing data.
    - Used to determine where to send the SMS by querying subscriber details.
4. **MSC (Mobile Switching Center)**:

    - A network element responsible for routing SMS messages to the **MS (Mobile Station)**, which is the recipient mobile device.
5. **MS (Mobile Station)**:
    - The mobile device (e.g., phone or modem) that is the recipient or sender of the SMS.

Message Flows

The communication between these entities follows the **SMPP protocol**, and the diagram highlights the sequence of commands and responses:

1. **submit_sm (Submit Short Message):**
    - The ESME sends a request to the SMSC (MC) to submit an SMS to a recipient.
    - The submit_sm is the SMPP command used for submitting the SMS.
2. **submit_sm_resp (Submit Short Message Response):**
    - The SMSC responds to the submit_sm command, acknowledging receipt of the request from the ESME.
    - This does not guarantee delivery to the end device yet—it only confirms the SMSC received the request.
3. **SRI_for_SM (Send Routing Information for Short Message):**
    - The SMSC queries the HLR to get routing information for the recipient's mobile device.
    - This ensures the SMS is routed to the correct MSC (where the recipient is currently registered).
4. **FSM (Forward Short Message):**
    - The SMSC forwards the SMS to the MSC for final delivery to the mobile device (MS).
    - The MSC will then deliver the message to the recipient.
5. **deliver_sm (Deliver Short Message):**
    - If the SMS delivery is successful, the SMSC sends a **delivery receipt** (DLR) back to the ESME using the deliver_sm command.
    - The DLR informs the ESME about the status of the sent message (e.g., delivered, failed, etc.).
6. **deliver_sm_resp (Deliver Short Message Response):**
    - The ESME sends a response to acknowledge the delivery receipt (deliver_sm) received from the SMSC.

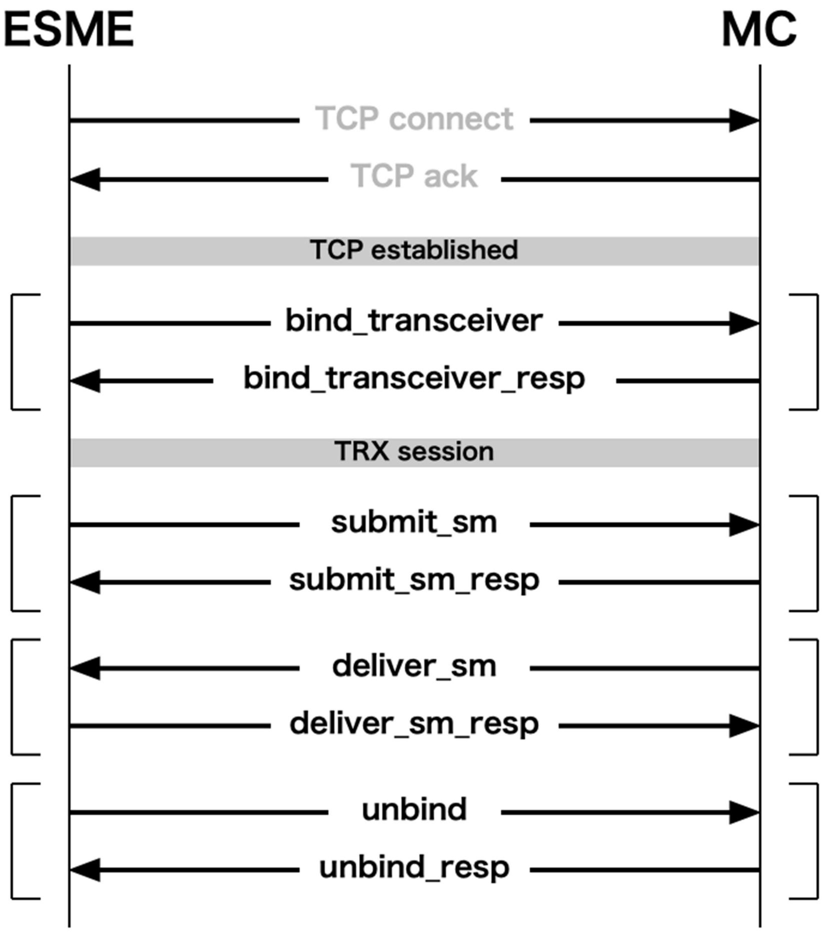
Key Points

- **SMPP**: It is a protocol used for exchanging SMS messages between ESMEs, SMSCs, and mobile networks.
- **DLR (Delivery Receipt)**: This is sent to the ESME to confirm message delivery status (success or failure).
- **HLR Query (SRI_for_SM)**: Ensures the SMSC can route the SMS correctly.
- **MSC Forwarding (FSM)**: This forwards the message to the actual mobile device.

Real-World Flow

- A notification application (ESME) sends an SMS request (submit_sm) to an SMS Gateway (SMSC).
- The SMSC queries the network (HLR) for routing info and forwards the message to the mobile network (MSC).
- Upon delivery, a delivery receipt (DLR) is generated (deliver_sm) and sent back to the application, confirming message delivery status.

This flow ensures a seamless exchange of messages and proper acknowledgment using the SMPP protocol.



This diagram illustrates the **SMPP (Short Message Peer-to-Peer) protocol** operations and the sequence of Protocol Data Units (PDUs) exchanged between an **ESME (External Short Message Entity)** and an **MC (Message Center/SMSC)**.

Overview of Diagram Components

1. **TCP Layer**:
   - The ESME and MC first establish a **TCP/IP connection** before exchanging SMPP PDUs.

    - The connection ensures the underlying communication channel is ready for message transfers.
2. **SMPP Bind Process**:
    - The ESME sends a **bind_transceiver** PDU to the MC.
    - The MC responds with a **bind_transceiver_resp** to confirm that the session has been successfully established.
    - This establishes a **TRX (Transceiver) session**, which allows **sending and receiving** messages over the same session.
3. **Submit Short Message (submit_sm)**:
    - Once the bind is complete, the ESME can send messages to the MC using the **submit_sm** PDU.
    - The MC responds with a **submit_sm_resp** PDU, indicating whether the message submission was successful or not.
        - If successful, it may include a **Message ID** for tracking.
4. **Delivery of Short Message (deliver_sm)**:
    - The MC sends a **deliver_sm** PDU to the ESME when a message (e.g., an incoming SMS or a delivery receipt) needs to be delivered.
    - The ESME responds with a **deliver_sm_resp** to acknowledge receipt of the message.
5. **Unbind Process**:
    - When the ESME wants to terminate the SMPP session, it sends an **unbind** PDU to the MC.
    - The MC confirms the termination with an **unbind_resp** PDU.
    - After this, the TCP connection can be closed.

Protocol Flow Steps

1. **TCP Establishment**:
    - The ESME connects to the MC using TCP/IP.
    - Once the connection is acknowledged, the session can begin.
2. **Bind Operation**:
    - The bind_transceiver command allows the ESME to both send and receive messages in a single session.
    - The response, bind_transceiver_resp, indicates successful binding.
3. **Message Submission**:
    - The ESME submits an SMS using the submit_sm PDU.
    - The MC confirms with a submit_sm_resp, providing the status.
4. **Message Delivery**:
    - When the MC delivers a message (such as an SMS or delivery receipt) to the ESME, it uses the deliver_sm PDU.
    - The ESME acknowledges the delivery with a deliver_sm_resp.
5. **Unbinding**:

    - To gracefully close the session, the ESME sends an unbind request.
    - The MC responds with an unbind_resp.

Key Points about SMPP PDUs

- Each operation (e.g., submit_sm, deliver_sm, unbind) consists of a **request PDU** and a **response PDU**.
- SMPP is session-based:
    - It requires a bind operation to start communication.
    - The session must be unbound before termination.
- The **TRX session** in this diagram allows bi-directional communication for sending and receiving messages within the same session.

Real-World Use Case

- A notification service (ESME) connects to an SMS Gateway (MC) to send SMS messages.
- It binds to the SMSC using bind_transceiver.
- It submits messages using submit_sm and receives delivery receipts (DLRs) via deliver_sm.
- After completing its work, it closes the session using unbind.

This process ensures robust, reliable communication between applications and SMS gateways using the SMPP protocol.