

SMPP 연동 규격서 정리

[Version 3.4 of the SMPP Protocol]

용어 정리 (규격서 확인)

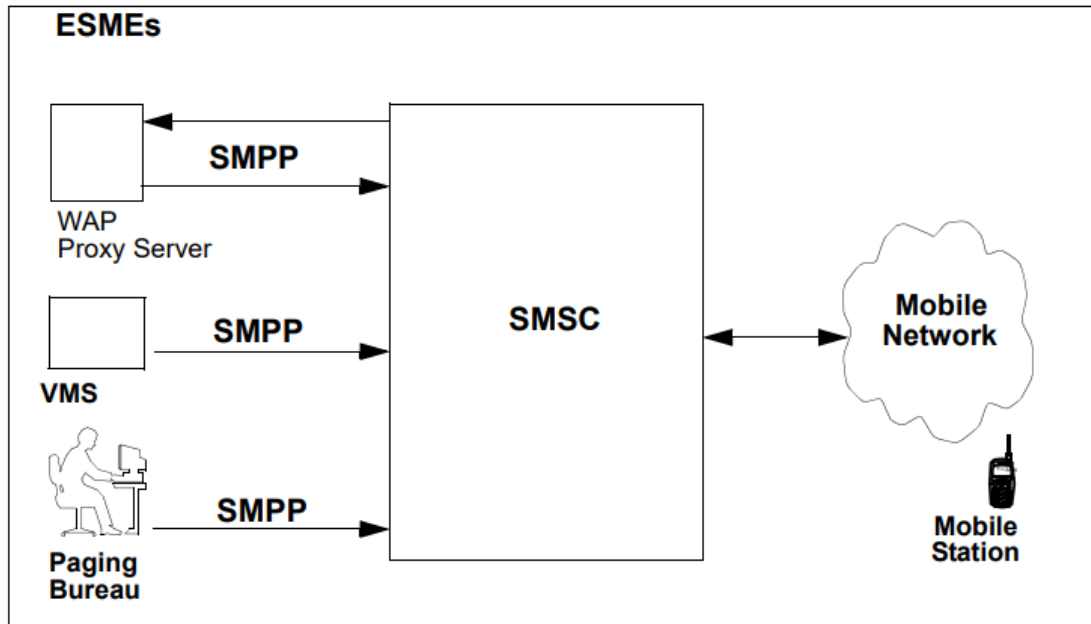
- SMSC (Short Message Service Centre)
- ESME (External Short Message Entity)
- PDU (Product Data Units)

1. SMPP Overview :

- SMPP (Short Message Peer to Peer) 국제 단문 메시지 표준 프로토콜이다.
- 간결함을 위해 SMPP Client == "ESME", SMPP Server == "SMSC"라고 한다.
- SMPP 프로토콜을 사용하는 것은 ESME 와 SMSC는 TCP/IP 통신으로 이루어진다.
- TCP/IP 통신을 통해 메시지 전송 및 수신이 가능하다.
- SMPP는 양방향 메시지 전송 기능을 지원한다.
- ESME에서 SMSC를 통해 단일 또는 여러 수신자로 메시지 전송이 가능
- SMPP (Short Message Peer to Peer) 국제 단문 메시지 표준 프로토콜이다.
- Message 전송 후 return 값으로 'delivery receipt'를 전달 받는다 SMSC로부터
- 메시지 전달 날짜 및 시간 설정이 가능 (예약 발송 기능)
- 메시지 모드를 선택 가능.
- 메시지의 전달 우선 순위 지정 가능.
- 메시지의 인코딩을 지정할 수 있다.
- 메시지의 유효 기간 설정 가능.

2. Scope :

- ESME ↔ SMSC Diagram



3. SMPP Protocol 정의 :

- a. ESME와 SMSC간에 단문 메시지를 교환하기 위한 작업을 말한다.
- b. SMPP 작업 중 ESME 애플리케이션이 SMSC와 교환해야 하는 데이터를 정의한다.
- c. SMS 지원 셀룰러 네트워크 가입자는 모바일에서 짧은 메시지를 수신할 수 있다
- d. SMPP는 요청 및 응답 프로토콜 데이터 유닛의(PDU) 교환을 기반으로 한다.
 - i. TCP/IP 네트워크 연결을 통한 ESME ↔ SMSC 통신
- e. 모든 SMPP 작업은 PDU와 연결된 응답 PDU로 구성되어야 한다.
- f. 수신 엔티티는 SMPP PDU 요청에 대한 연관된 SMPP 응답을 반환해야 한다.
- g. 이 규칙의 예외는 **alert_notification** PDU로 응답이 없다.
- h. ESME와 SMSC간의 SMPP를 통한 메시지 교환은 아래와 같이 세 가지 타입으로 분류될 수 있다.
 - i. **Transmitter** : ESME → SMSC로 보낸 메시지를 말한다.
 - ii. **Receiver** : SMSC → ESME로 보낸 메시지를 말한다.
 - iii. **Transceiver** : ESMS ↔ SMSC로 요청 및 응답을 받을 수 있다.

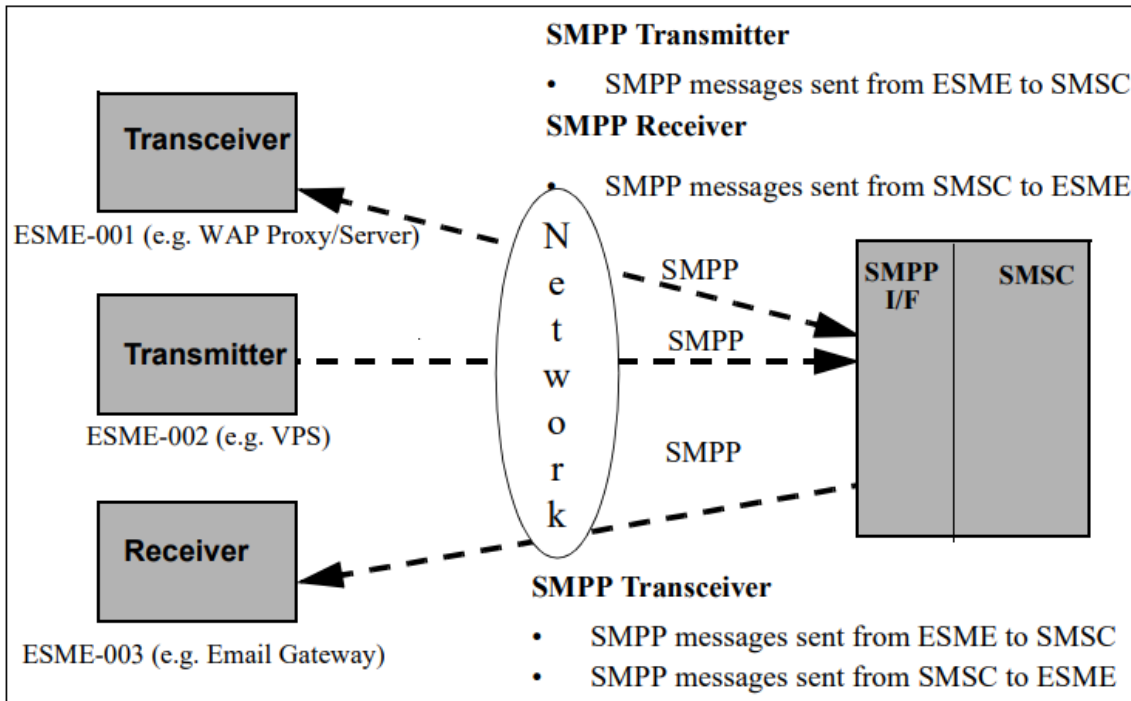


Figure 2-1: SMPP interface between SMSC and ESME

4. SMPP Session Description

- 네트워크 설정:** SMSC와 ESME간의 SMPP 세션은 ESME가 먼저 SMSC와 네트워크 연결을 설정한 후 SMPP 바인드 요청을 발행하여 SMPP 세션을 열도록 한다.
- ESME가 메시지를 제출하고 받기를 원하는 경우 두 개의 네트워크 연결 (TCP/IP) 및 두 개의 SMPP 세션을 설정해야 한다. **(Transmitter and Receiver)**
- 해당 버전에서는 대안으로 단일 네트워크 연결을 통해 SMPP 송수신 세션을 설정할 수 있다. **(Transceiver)**
- The SMPP session may be defined in terms of the following possible states:**

i. **OPEN (Connected and Bind Pending) :**

- ESME가 SMSC에 대한 네트워크 연결을 설정했지만 아직 바인드를 하지 않은 상태

ii. **BOUND_TX:**

- 연결된 ESME가 ESME 송신기로 바인딩 하도록 요청했고, SMSC로부터 바인드 요청이 승인됨.
- 이 상태는 ESME → SMSC로 단문 메시지를 보낼 수 있음.

iii. **BOUND_RX:**

1. 연결된 ESME가 ESME 수신기로 바인딩 하도록 요청했고, SMSC로부터 바인드 요청이 승인됨.
2. 이 상태는 ESME \leftarrow SMSC로 단문 메시지를 받을 수 있음.

iv. **BOUND_TRX:**

1. 연결된 ESME가 ESME 송수신기로 바인딩 하도록 요청했고, SMSC로부터 바인드 요청이 승인됨.
2. 이 상태는 ESME \leftrightarrow SMSC 송신기와 수신기 기능을 모두 지원

v. **CLOSED (Unbound and Disconnected):**

1. ESME가 SMSC와의 바인드를 해제하고 네트워크 연결을 종료한 상태이다.

5. **Outbind**

- a. **목적** : outbind 작업의 목적은 SMSC가 ESME에게 메시지를 전달하기 위해 bind_receiver 요청을 보낼 수 있도록 하는 것이다. 예를 들어, SMSC가 아직 ESME로 전달되지 않은 메시지를 가지고 있을 때 유효하다.
- b. **시작** : outbind SMPP 세션은 SMSC가 먼저 ESME에 대한 네트워크 연결을 설정한 후에 시작한다.
- c. **프로세스**: 네트워크 연결이 설정된 후, SMSC는 ESME에게 "outbind" 요청을 보내어 바인드한다. ESME는 "bind_receiver" 요청으로 응답해야 하고, 이에 대한 SMSC의 응답은 "bind_receiver_resp"이다.
- d. **거절**: ESME가 outbind 세션을 수락하지 않는 경우(예: 잘못된 system_id 또는 비밀번호 등), ESME는 네트워크 연결을 해제해야 한다.
- e. **특성**: SMPP 세션이 한 번 설정되면 이는 일반적인 SMPP 수신기 세션과 동일한 특성을 가진다.

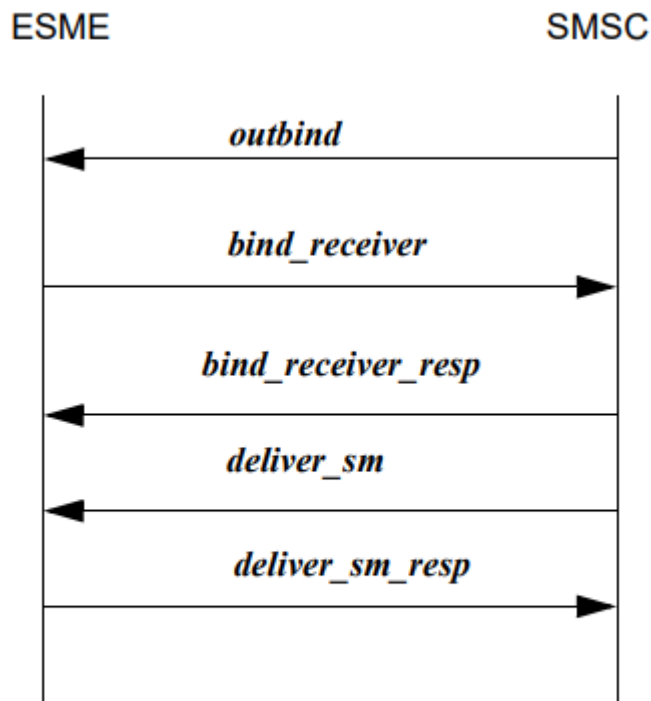


Figure 2-2: Sample Outbind Sequence

6. SMPP messages sent from ESME to SMSC

- ESME는 메시지를 보내기 위해 ESME 송신기 또는 송수신기로 SMSC로 연결되어 있어야 한다.
- ESME 송신기에서 SMSC로 전송되는 SMPP 메시지 유형에는 (`submit_sm`)과 (`data_sm`) 포함된다.
- ESME는 SMSC에게 이전에 제출한 메시지의 상태를 조회 하거나(`query_sm`), 이전에 제출한 메시지의 전송을 취소하거나(`cancel_sm`), 이전에 제출한 메시지를 교체하는(`replace_sm`) 등의 SMPP 작업을 수행할 수 있다.
- ESME가 SMSC에게 전송한 SMPP PDU는 SMSC에 의해 수신 되었을 때 PDU 응답을 줘야한다.

7. SMPP Message Response from SMSC to ESME

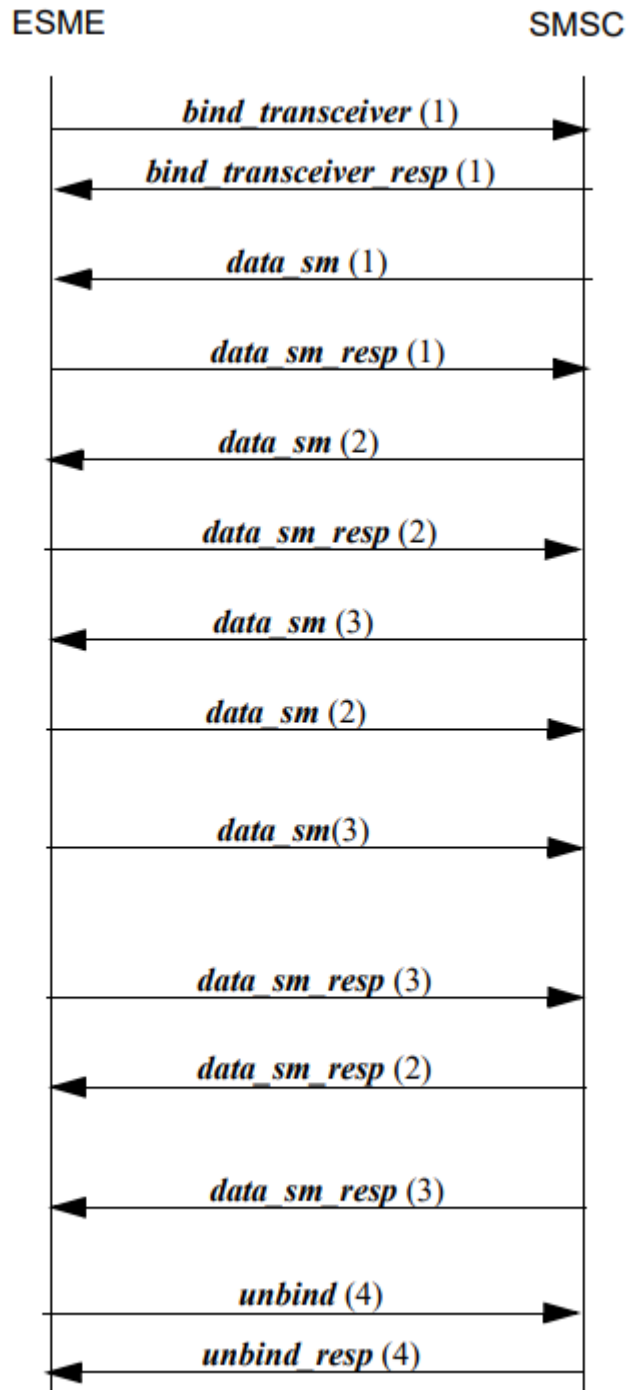
- SMM PDU 응답에는 해당 메시지에 할당된 고유한 식별자가 포함된다. (해당 메시지 식별)

- b. 응답에는 제출된 메시지가 유효한지 아닌 지를 나타내는 상태가 포함된다. 아닐 경우 SMSC는 적절한 오류 상태를 return

c. **응답 PDU**

- i. `submit_sm_resp`
- ii. `data_sm_resp`
- iii. `query_sm_resp`
- iv. `cancel_sm_resp`
- v. `replace_sm_resp`

8. **Typical SMPP session sequence - ESME Transceiver** (내가 구현할 BIND Type)



- 동기식 및 비동기식 구현이 가능하다. 따라서 위에 사진처럼 SMSC는 연속적인 data_sm 요청을 ESME에게 비동기적으로 보낼 수 있다.
- 비동기식 요청 및 응답은 SMSC가 비동기적으로 발행한 연속적인 data_sm 요청은 ESME가 연관된 PDU로 응답을 빠르게 해야 한다. 이 때 SMPP 헤더의 sequence_number 매개변수는 SMPP 응답 PDU를 SMPP 요청 요청 PDU와 연결 하기 위해 사용

- 응답 순서 유지를 해야 한다. ESME는 항상 원래 요청이 받은 순서대로 SMPP 응답을 SMSC에게 반환해야 한다. 그러나 이는 SMPP에서 의무적인 사항은 아니며, SMSC는 순서가 바뀐 응답을 처리할 수 있어야 한다.
- EMSE ↔ SMSC간 최대 대기 중인 작업 수에 대한 제한은 없지만, 권장 사항은 10개 이하다.

9. SMPP Error Handling

- 모든 SMPP 작업은 요청 PDU와 연관된 응답 PDU로 구성된다.
(`alert_notification`) PDU를 제외하고는 모든 경우에 대응하는 SMPP 응답 PDU를 반환해야 한다.
- SMPP 요청 PDU에 오류가 있으면, 수신 측은 응답 PDU 헤더의 `command_status` 필드에 적절한 오류 코드를 삽입하여 응답을 반환해야 한다.
- PDU 헤더에 오류가 있는 경우, 수신 측은 originator에게 `generic_nak` PDU를 반환해야 한다.

10. SMPP Timers

- 효율적인 SMPP 트랜잭션 교환을 보장하기 위해 각 SMPP 세션은 ESME와 SMSC 간의 통신 SMPP 엔티티에서 구성 가능한 타이머를 사용해서 관리해야 한다.
- 아래와 같은 SMPP 타이머를 설정하는 것이 권장한다.
 - **SMPP 세션 시작 타이머:** ESME가 SMPP 세션을 시작할 때, 네트워크 연결을 열고 일정 기간 내에 이를 수행하는 것을 보장한다.
 - **SMPP 세션 타이머:** ESME 또는 SMSC가 다른 통신 SMPP 엔티티의 SMPP 세션 상태를 `enquire_link` 명령을 통해 요청할 수 있도록 한다.
 - **SMPP 비활동 타이머:** SMPP 메시지가 교환되지 않는 경우 최대 기간을 지정하고, 해당 시간이 경과하면 SMPP 세션을 정상적으로 종료한다.
 - **SMPP 트랜잭션 타이머:** SMPP 요청과 해당하는 SMPP 응답 사이의 경과 시간을 지정합니다.

11. SMPP Message Modes

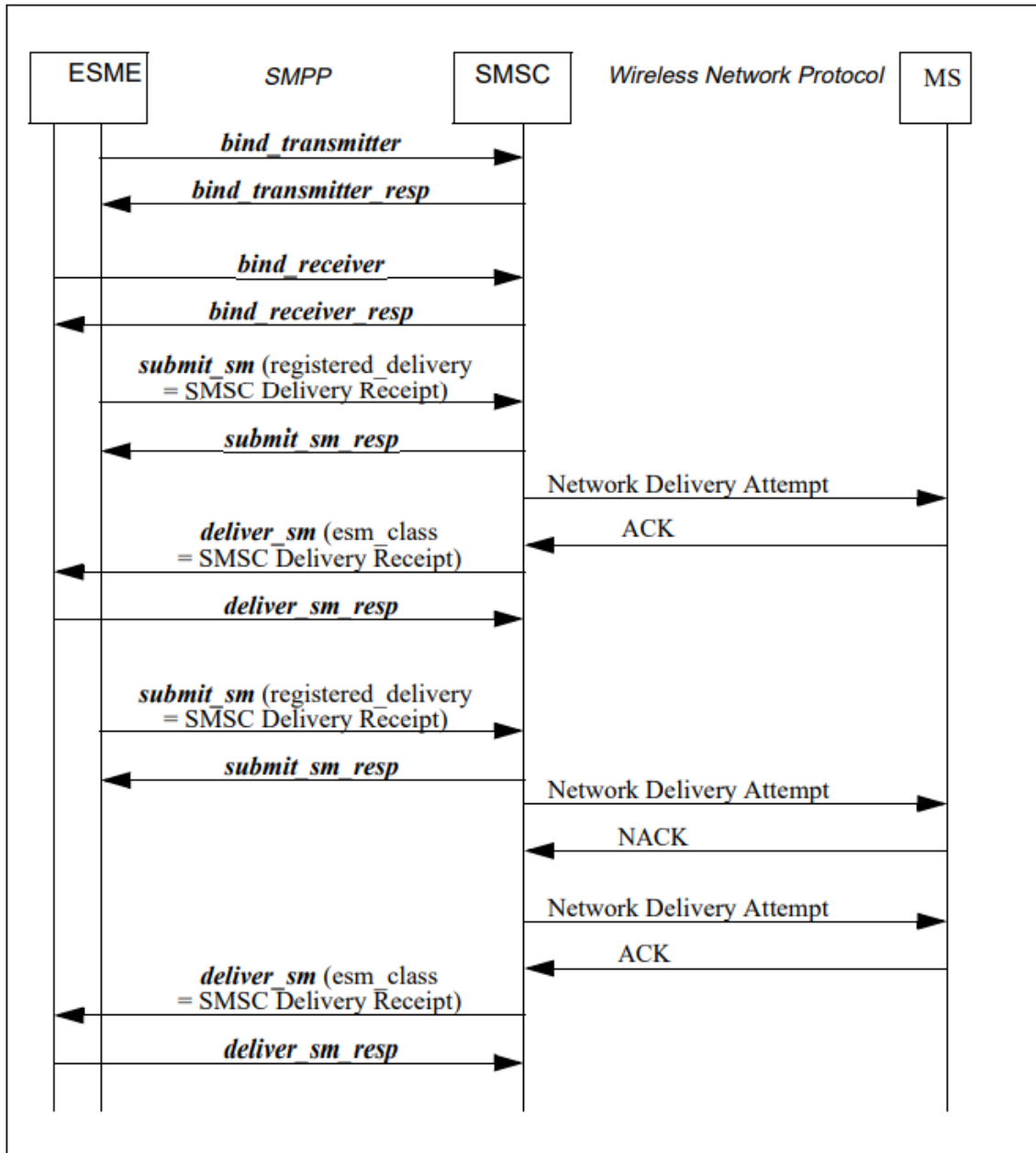
- SMPP는 SMSC에서 지원되는 경우 ESME가 SMSC 메시지 전달 메커니즘을 선택할 수 있는 메시지 모드 옵션을 제공한다.
- 일반적으로 SMSC에서 제공할 수 있는 전달 메커니즘은 아래와 같다.
 - Store and Forward** (저장 및 전달)

ii. **Datagram** (데이터그램)

iii. **Transaction mode** (트랜잭션 모드)

12. **Store and Forward Message Mode** (내가 쓸 메시지 모드)

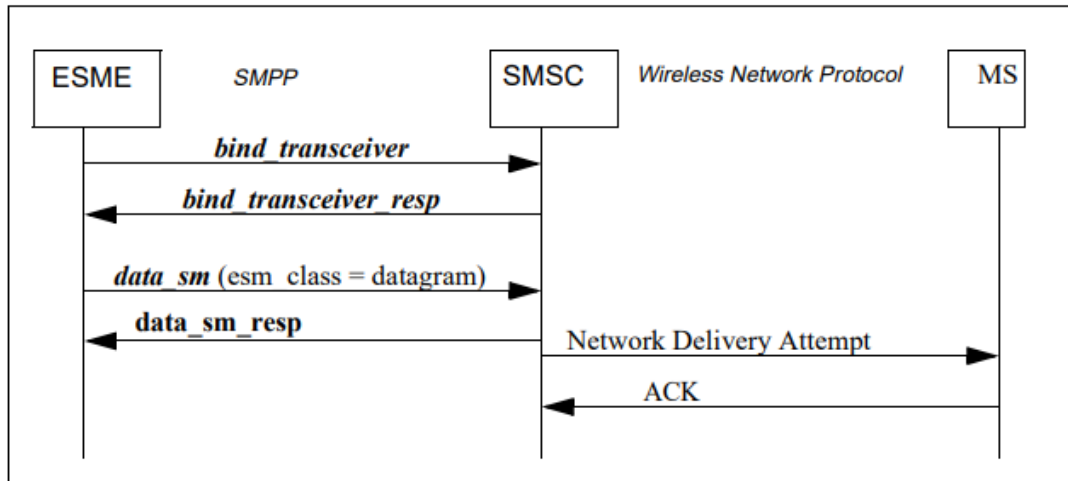
- a. SMS의 접근 방식은 메시지를 수신 SME에게 전달하기 전 SMSC 저장 영역 (메시지 DB)에 메시지를 저장했었다.
- b. 이 모델에서는 메시지가 SMSC에 의해 모든 전달 시도가 완료될 때까지 안전하게 저장된다.
- c. SMPP는 `submit_sm` 작업을 통해 저장 및 전달 메커니즘을 지원한다.
- d. 이를 통해 ESME는 메시지를 SMSC로 보내고 성공적으로 전달되거나 메시지 유효기간이 만료될 때까지 저장되고, `data_sm` 작업을 통해서도 지원된다.
- e. "저장 및 전달" 메시지 모드는 저장된 단문 메시지에 대한 이후 SMPP 작업을 용이하게 하고, `query_sm`, `replace_sm`, `cancel_sm` 과 같은 작업을 통해 저장된 단문 메시지에 대한 후속 SMPP 작업이 지원된다.
- f. SMS 전달의 최종 결과를 결정하려면 ESME가 `submit_sm` 또는 `data_sm` 작업에서 SMSC 배달 영수증을 요청해야 한다.



13. Datagram Message Mode

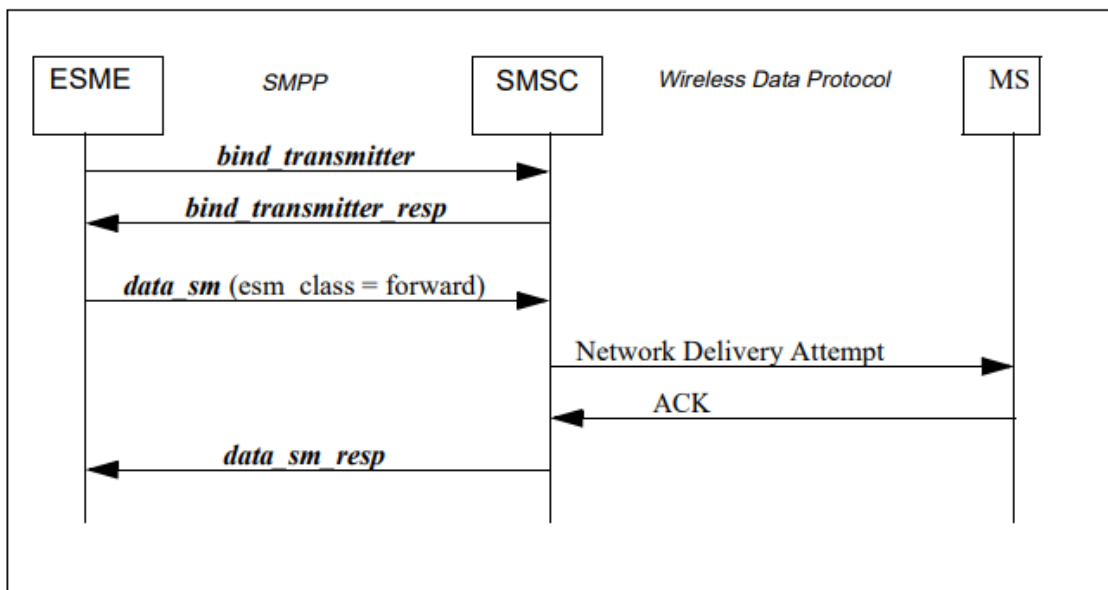
- 메시지 안정 저장 및 재시도 보장을 포함하지 않고 높은 메시지 처리량에 초점을 맞춘다.
- 예약된 전달, 등록된 전달 등과 같은 일반적인 SMSC 기능은 적용되지 않는다.
- 데이터그램 메시지 모드는 저장 및 전달 메시지 모드에서 제공되는 높은 수준의 안전한 전송 기능이 필요하지 않고, 고 처리량 애플리케이션을 위해서 설계되었다.
- 주로 데이터 콘텐츠, 차량 추적 애플리케이션 등이 해당
- 데이터그램 모드는 data_sm 작업을 통해 지원한다.

f. esm_class 매개변수를 사용하여 Datagram 메시지 모드를 선택한다.



14. Transaction Message Mode

- ESME 메시지 발신자가 SMPP 응답 PDU 내에서 메시지가 목적지 MS로 성공적으로 또는 실패로 전달 되었는지를 나타내는 형태의 전달 확인을 받을 수 있도록 한다.
- 트랜잭션 메시지 모드는 ESME가 장기간의 SMSC 저장이 필요하지 않은 동기식 종단 간 전달 결과를 필요로 하는 실시간 메시징을 포함하는 애플리케이션을 위한 설계 되었다.
- SMPP는 트랜잭션 메시지 모드를 **data_sm** esm_class 매개변수를 사용하여 트랜잭션 메시지 모드를 선택한다.



15. Message Types

a. 메시지 유형은 위 SMPP 작업의 esm_class 매개변수에 정의된다.

b. SMSC Delivery Receipt

- i. 이 메시지 유형은 SMSC 전달 영수증을 전달하는 데 사용됩니다.
- ii. SMSC는 SMSC에 저장된 등록 메시지의 최종 상태를 감지하면 메시지의 발신자에게 주소 지정된 영수증 메시지를 생성해야 한다.
- iii. SMSC 전달 영수증은 SMPP deliver_sm 또는 data_sm 작업의 사용자 데이터 페이로드로 전송된다.
- iv. 전달 영수증을 전송하는 데 사용되는 deliver_sm 및 data_sm 작업에서 다음 필드가 관련이 있다.

v. **출발지 주소 (source address):**

- 1. 출발지 주소는 원래의 짧은 메시지의 목적지 주소에서 가져온다.
- 2. 출발지 주소는 "source_addr_ton" (출발지 주소 유형), "source_addr_npi" (출발지 주소 번호 계획 식별자), "source_addr" (출발지 주소) 세 가지 필드로 구성된다.
- 3. 즉, 배달 영수증을 생성한 원본 짧은 메시지의 목적지 주소를 출발지 주소로 사용한다.

vi. **목적지 주소 (destination address):**

- 1. 목적지 주소는 원래의 짧은 메시지의 출발지 주소에서 가져온다.
- 2. 목적지 주소는 "dest_addr_ton" (목적지 주소 유형), "dest_addr_npi" (목적지 주소 번호 계획 식별자), "destination_addr" (목적지 주소) 세 가지 필드로 구성된다.
- 3. 즉, 배달 영수증을 생성한 원본 짧은 메시지의 출발지 주소를 목적지 주소로 사용한다.

vii. esm_class

viii. message_state

ix. network_error_code

x. receipted_message_id

16. SMPP PDU - Type Definitions

a. Integer (Unsigned value with the defined number of octets):

- i. 이는 부호 없는 정수를 의미합니다. "정해진 수의 옥텟(octets)"으로 구성되어 있으며, 옥텟은 8비트 바이트를 말한다.
- ii. 따라서, 이 정수는 8비트 단위로 구성됩니다."MSB first (Big Endian)"는 가장 중요한 비트(Most Significant Bit) 또는 가장 큰 바이트가 먼저 전송되는 것을 의미한다.

b. C-Octet String:

- i. ASCII 문자의 시리즈로 구성되며, NULL 문자(즉, 값이 0인 문자)로 종료됩니다. 이는 문자열의 끝을 나타내기 위해 사용됩니다.
- ii. 예를 들어, "Hello" 문자열은 'H', 'e', 'l', 'l', 'o', '\0'으로 표현됩니다.

c. C-Octet String (Decimal):

- i. ASCII 문자의 시리즈로 구성되며, 각 문자는 10진수 숫자(0-9)를 나타냅니다. 이 문자열도 NULL 문자로 종료된다.
- ii. 예를 들어, 숫자 123은 '1', '2', '3', '\0'으로 표현된다.

d. C-Octet String (Hex):

- i. ASCII 문자의 시리즈로 구성되며, 각 문자는 16진수 숫자(0-F)를 나타냅니다.
- ii. 이 문자열도 NULL 문자로 종료된다.
- iii. 예를 들어, 16진수 값 1F3은 '1', 'F', '3', '\0'으로 표현된다.

e. Octet String:

- i. 옥텟의 시리즈로 구성되며, 반드시 NULL로 종료되지는 않는다.
- ii. 이는 바이너리 데이터를 전송할 때 사용되며, 문자열이 아닌 데이터(예: 이미지, 비디오 클립 등)를 나타낼 때 사용된다.

f. Notes

i. Octet-String 필드의 NULL 설정에 대한 참조:

- 1. 이는 해당 필드가 단일 NULL 문자로 구성되어 있음을 의미한다.
- 2. 즉, 값이 0x00(0)으로 인코딩된 하나의 옥텟이다.
- 3. NULL 문자는 데이터의 끝을 나타내는 데 일반적으로 사용되며, 여기서는 그것이 단일 옥텟으로 표현되어 있음을 나타낸다.

ii. 정수(Integer) 필드의 NULL 설정에 대한 참조:

- 1. 이는 해당 필드가 0으로 채워짐을 의미한다.

2. 여기서 "NULL 설정"이라는 용어는 실제로 정수 필드가 0값으로 초기화되어 있음을 나타낸다.

3. 이는 특정 필드가 비어 있거나 기본값이 없음을 표현할 때 사용될 수 있다.

iii. 모든 **C-Octet String** 형식의 경우, 최대 필드 크기:

iv. 최대 필드 크기는 문자열 길이와 NULL 종결자를 포함한 조합으로 표시된다.

v. 예를 들어, 8문자로 구성된 C-Octet String은 NULL 종결자가 포함될 때 9개의 옥텟으로 인코딩된다.

vi. 이는 문자열의 실제 내용을 나타내는 옥텟들과 문자열의 끝을 나타내는 NULL 문자(즉, 0x00으로 인코딩된 옥텟)를 포함하여 계산된다.

vii. 따라서, 문자열의 최대 길이를 계산할 때는 항상 NULL 종결자를 고려해야 한다.

17. SMPP Parameter Field Size Notation

Size octets	Type	Description of String type specified
4	Integer	고정 크기의 정수 필드이다. 예시에서는 32비트(4 옥텟) 크기의 정수를 나타낸다.
Var Max 16	C-Octet String (Dem	이 문자열은 1-15 ASCII 문자로 구성된 가변 길이를 가지며, NULL 종결자를 포함하는 추가 옥텟에 의해 종료된다. 빈 문자열은 NULL 문자(0x00)를 포함하는 단일 옥텟으로 인코딩된다.
Fixed 1 or 17	C-Octet String	NULL 문자를 포함하는 1 옥텟 또는 고정된 수의 문자(예시에서는 16문자) 뒤에

		NULL 문자가 오는 구조다.
Var 0 - 254	Octet String	가변 크기의 옥텟 문자열 필드다. 예시에서는 크기가 0에서 254 옥텟까지 변할 수 있다.

18. SMPP PDU Header

SMPP PDU				
PDU Header (mandatory)				PDU Body (Optional)
<i>command length</i>	<i>command id</i>	<i>command status</i>	<i>sequence number</i>	<i>PDU Body</i>
4 octets	Length = (Command Length value - 4) octets			

Table 3-2: SMPP PDU Format Overview

- **command_length (4 바이트 정수):**
 - SMPP PDU 패킷의 전체 옥텟 길이를 정의.
 - 이 길이에는 길이 필드 자체도 포함된다.
- **command_id (4 바이트 정수):**
 - 특정 SMPP PDU를 식별 (예: `submit_sm`, `query_sm` 등).
 - 각 SMPP 요청 PDU에는 0x00000000에서 0x000001FF (**0 ~ 511**) 범위 내에서 고유한 명령 식별자가 할당.
 - 각 SMPP 응답 PDU에도 0x80000000에서 0x800001FF 범위 내에서 고유한 명령 식별자가 할당. (응답 SMPP command_id는 해당 요청 SMPP command_id와 동일하지만 31번째 비트가 설정됨).
- **command_status (4 바이트 정수):**
 - SMPP 요청의 성공 또는 실패를 나타냄.
 - 이 필드는 SMPP 응답 PDU에서만 관련이 있으며, SMPP 요청 PDU에서는 NULL 값을 포함해야 한다.
- **sequence_number (4 바이트 정수):**
 - SMPP 요청과 응답을 상관 시키기 위해 사용되는 일련 번호를 포함한다.

- 메시지 상관 관계를 위해 일련 번호를 사용하면 SMPP PDU가 비동기적으로 교환 될 수 있습니다.
- sequence_number의 할당은 SMPP PDU의 발신자가 책임진다.
- 제출된 각 SMPP 요청 PDU에 대해 sequence_number는 단조롭게 증가해야 하며, 관련 SMPP 응답 PDU에서 보존되어야 합니다.
- sequence_number는 0x00000001에서 0x7FFFFFFF (1 ~ **2147483647**)범위일 수 있습니다.

19. SMPP PDU Length

a. command_length 필드:

- SMPP PDU 헤더의 맨 처음에 위치하며, 해당 SMPP PDU에 포함된 총 옥텟 (바이트) 수를 나타낸다.
 - command_length 필드는 4옥텟 정수로, Big Endian 형식으로 전송된다.
- PDU 길이 해석:** ESME 또는 SMSC는 SMPP PDU를 해석할 때, 먼저 command_length 필드(4옥텟)를 읽어 PDU의 길이를 결정한다.
 - 그런 다음, command_length 필드의 길이(4옥텟)를 전체 PDU 길이에서 빼서 남은 데이터의 양을 결정한다.
 - 따라서 N값의 command_length를 추출하면, 주어진 PDU에 대해 N-4 옥텟이 남아 있음을 나타냅니다.

Example:-

The following data-stream example illustrates how the SMPP PDU header is encoded:

```
00 00 00 2F 00 00 00 02 00 00 00 00 00 00 00 01 53 4D 50 50 33 54 45 53 54 00
73 65 63 72 65 74 30 38 00 53 55 42 4D 49 54 31 00 00 01 01 00
```

Note: Values are shown in Hex format.

The header would be decoded as follows:

00 00 00 2F	Command Length	0x0000002F
00 00 00 02	Command ID	0x00000002 (<i>bind_transmitter</i>)
00 00 00 00	Command Status	0x00000000
00 00 00 01	Sequence Number	0x00000001

The remaining data represents the PDU body (which in this example relates to the *bind_transmitter* PDU).

20. SMPP Message length and extended message length

a. sm_length 필드:

- i. `submit_sm`, `submit_multi`, `deliver_sm`, `replace_sm` SMPP PDU에서 메시지 텍스트(또는 사용자 데이터)의 길이를 정의한다.
- ii. `sm_length` 필드에서 지정할 수 있는 최대 메시지 길이는 254 옥텟이다.

b. 254 옥텟을 초과하는 메시지:

- i. ESME가 254 옥텟을 초과하는 메시지를 전송하려는 경우, `sm_length` 필드를 NULL로 설정해야 하며, `message_payload` Optional 매개변수를 사용하여 메시지 길이 값과 사용자 데이터를 포함해야 한다.

c. 확장 메시지 길이 지원:

- i. SMPP는 `submit_sm`, `submit_multi`, `data_sm`, `deliver_sm` PDU에서 확장 메시지 길이를 지원한다.
- ii. 이는 메시지가 254 옥텟을 초과할 수 있음을 의미한다.

21. "BIND" Operation

- a. 메시지 전송 전에 ESME → SMSC 바인드 요청을 해야 한다. (인증 및 연결 과정)
- b. 이 연결을 통해서 메시지를 전송할 수 있다.
- c. 위에 설명한 것과 같이 바인드 종류는 3가지가 존재한다.
 - i. Transmitter (ESME → SMSC 송신 가능) **SMSC -- ESME**
 - ii. Reciver (ESME ← SMSC 수신 가능) **SMSC ← ESME**
 - iii. Transceiver (ESME ↔ SMSC 송수신 가능) **SMSC ↔ ESME**

d. UNBIND

- i. BIND 해제 요청이 온다. SMSC → ESME 인스턴스를 해제

e. UNBIND_RESP

- i. 위에 대한 응답

22. "GENERIC_NACK" (NEGATIVE ACKNOWLEDGEMENT) **SMSC → ESME**

- a. 유효하지 않은 메시지 헤더를 보내거나
- b. `content_length`가 유효하지 않거나 (너무 짧거나, 너무 긴)
- c. 알 수 없거나 유효하지 않은 `command_id`를 보낸 경우

- d. 위에 3가지 케이스일 경우 **GENERIC_NACK**가 온다.
- e. **generic_nack** 응답은 특정한 오류 코드나 상세한 오류 정보 없이 일반적인 오류 응답으로 사용된다.

밑에 전송 및 리포트는 내용이 많기에 규격서를 참조할 수 있도록 한다.

23. "SUBMIT_SM" (간단하게 설명) ESME → SMSC

- a. 트랜잭션 메시지 모드를 지원하지 않는다.
- b. 이름과 같이 SMSC로 전송한다.
- c. Header **sequence_number** 를 전송할 때 보냈던 값을 submit_resp에서 그대로 돌려 받는다.
 - i. 비동기로 전송했을 경우 해당 **sequence_number** 를 가지고 내가 보냈던 메시지를 확인할 수 있다.
- d. **schedule_delivery_time** , **validity_period**
 - i. SMSC 시간으로부터 절대 시간, 상대 시간을 정의할 수 있다.
 - ii. 시간 관련 포맷은 규격서 SMPP 3.4 Version 7.1.1 Section을 참조하면 된다.
 - iii. **schedule_delivery_time** 는 예약된 시간을 지정하여, 예약 발송을 하기 위한 필드다. 시간을 지정하기 위해서는 **타임 Section** 참조하고, 즉시 전달을 원한다면 null로 설정한다.
 - iv. **validity_period** 는 SMSC 만료 시간을 나타낸다. 해당 만료 시간 동안 메시지가 도착하지 않을 경우 폐기할 수 있으며, 이것 또한 **타임 Section** 참조하여 설정할 수 있다. null로 설정하면 기본 유효 기간이다.
- e. **registered_delivery**
 - i. SMSC에 전송된 메시지의 결과를 확인을 요청하는 데 사용된다.
 - ii. 이 필드를 통해 메시지 전송의 결과를 알려주는 다양한 옵션을 제공하여, 메시지가 성공적으로 전달되었는지, 실패했는지 등의 정보를 제공 받을 수 있다.
 - iii. **0x01** 값을 넣으면 최종 목적지에 도달하였는지, 실패했는지에 대한 배달 영수증을 받을 수 있다. 해당 필드에 이 값을 넣어줘야 한다.
- f. **replace_if_present_flag**
 - i. 매개변수는 SMSC(Short Message Service Center)에게 이전에 제출되었지만 아직 배달되지 않은 메시지를 새 메시지로 대체하도록 요청하는 데 사용된다.

ii. SMSC는 원본 주소(source address), 목적지 주소(destination address), 그리고 서비스 유형(service_type)이 새 메시지와 같은 경우에 기존 메시지를 대체한다.

iii. 이 매개변수의 값은 아래를 참조:

1. 0 : 대체하지 않음 (기본값), 1 : 대체, 2 - 255 : 예약됨
2. 1 을 설정할 경우 service_type을 사용해야 한다.
3. 대체를 할 경우 (동일한 발신자, 수신자, 서비스 유형)이 충족될 때 이전에 발송된 건이 아직 도착하지 않은 메시지를 새로운 메시지로 대체하도록 SMSC에 요청하는 데 사용이 된다.

g. data_coding

- i. 인코딩 체계를 정의한다.
- ii. GSM-7 사용하면 메시지를 최대 160자까지 보낼 수 있기에 GSM-7로 한다.
- iii. 0x00 값을 넣으면 **GSM-7 Or Default Alphabet**이라고 불리는 인코딩으로 하며 1octet이 8비트가 아닌 7비트이다. 자세한 설명은 아래 따로 **GSM-7** Section 참조

h. sm_default_msg_id

- i. 사용하지 않는데, 혹시 모르니 간단하게 말을 하면 반복적으로 사용되는 메시지 예를 들어 회원가입을 축하합니다. 같은 메시지를 미리 SMSC에 1이란 번호로 저장하여, 이 필드에 1이란 값을 넣어서 보내면 해당 문자 내용으로 나가는 것이다.
- ii. 알림톡, RCS 템플릿과 비슷한 개념이라고 생각하면 된다.

i. priority_flag

- i. 메시지의 우선순위를 지정한다.
- ii. 0 = 레벨 0 (가장 낮은) 우선 순위
- iii. 1 = 레벨 1 우선 순위
- iv. 2 = 레벨 2 우선 순위
- v. 3 = 레벨 3 (가장 높은) 우선 순위

24. "SUBMIT_SM_RESP" (간단하게 설명) SMSC ← ESME (알아서 확인)

25. "DELIVER_SM" (Report)

- a. **SUBMIT_SM** 전송 시에 **registered_delivery** 해당 deliver_sm을 받도록 설정을 하였으면 DELIVER_SM (Report)이 온다.
- b. 이제 deliver_sm에서 확인해야 하는 필드는 short_message이다.
- c. SMPP 3.4 version "DELIVER_SM" Operation 부분을 보면 Reference Appendix B for an example Delivery Receipt format. 이렇게 나와있다.
- d. 그러면 예시 format을 보고 개발을 진행하면 된다.
- e. **Appendix B** 부분을 참조하면 쉽게 개발이 가능하다.

```
"id:IIIIIIII sub:SSS dlvr:DDD submit date:YYMMDDhhmm done
date:YYMMDDhhmm stat:DDDDDDD err:E Text: ....."
```

The fields of the above delivery receipt example are explained in the following table:

26. "DELIVER_SM_RESP" (알아서 확인)

27. GSM-7 (Global System for Mobile Communications)

- a. 네트워크에서 사용되는 표준 문자 인코딩 방식이다.
- b. GSM 7 비트 인코딩은 최대한의 문자 메시지 길이를 확보하기 위해 설계 되었으며, 한 문자를 표현하는 데 8비트가 아닌 7비트를 사용한다.
- c. **GSM-7 특징**
 - i. **문자 크기:** GSM 7비트 인코딩은 이름에서 알 수 있듯이, 한 문자를 표현하는 데 7비트를 사용. 이는 8비트를 사용하는 ASCII 인코딩이나 16비트를 사용하는 유니코드(UTF-16) 인코딩과 대비된다.
 - ii. **문자 수용량:** 7비트 인코딩을 사용함으로써, 표준 SMS 하나에 최대 **160개의 문자**를 포함할 수 있습니다. 만약 다른 인코딩 방식을 사용한다면, 문자당 더 많은 비트가 필요하기 때문에, SMS 당 포함할 수 있는 문자 수가 줄어든다.
 1. Ex ⇒ **UCS2** (70), **Latin** (140), **Unicode** (70)
 - iii. **지원 문자 세트:** GSM 7비트 기본 알파벳은 영문 대소문자, 숫자, 일반적인 구두점 및 특수 문자, 그리고 유럽 언어의 일부 특수 문자를 포함.
 - iv. **확장 문자 세트:** 특정 기호와 문자를 포함하기 위해, GSM 7비트 인코딩은 확장 문자 세트도 제공한다. 이를 위해 이스케이프 문자(ESC, 0x1B)를 사용하며, 이는 일반 문자 세트에 없는 문자를 나타내는 데 사용된다. 그러나, 확장 문자를 사용할 때는 두 개의 7비트(즉, 14비트)가 필요해져서, 이러한 문자를 포함할 경우 SMS 당 문자 수가 줄어듭니다.

28. Time Format

- Format : `YYMMDDhhmmsstnnp`
- t는 1초의 10분의 1 (0-9)
- nn은 현지 시간과 UTC 사이의 시간 차이를 15분 단위로 나타낸다(00-48)
- p는 시간 차이의 방향을 나타내며, +는 현지 시간이 UTC보다 앞서는 경우, -는 현지 시간이 UTC 보다 늦는 경우다.
- **Absolute Time Foramt**
 - 예정된 전달 시간과 만료 시간을 세계적인 UTC 형식으로 지정한다. 이것을 15분 단위의 오프셋과 방향 기호 + 또는 - 포함
- **Relative Time Format**
 - UTC 방향 플래그를 '+' 또는 '-' 대신 'R'로 설정함으로써 표시될 수 있다
 - 상대 시간은 1초의 10분의 1을 나타내는 't'와 UTC 오프셋 'nn'은 무시되며 각각 '0'과 '00'으로 설정해야 한다
 - Ex ⇒ **020610233429000R** 현재 SMSC 시간으로부터 2년, 6개월, 10일, 23시간, 34분, 29초의 상대 기간으로 해석된다.