

2016 도로교통분야 ITS 표준 및 성능평가 정기교육

**기술기준의 이해 및 적용을  
위한 심화교육**

2016. 10.

# - 목 차 -

<b>제1장 기술기준 이해</b>	<b>1</b>
1. 개요	1
2. 기본교통정보교환 기술기준 I	2
2.1 개요	2
2.2 기술기준 구성 및 적용범위	2
2.3 정보의 구성	3
2.4 교통정보세항목 구현방법	4
2.4.1 교통소통정보(101)	4
2.4.2 교통통제정보(102)	5
2.4.3 돌발상황발생정보(103)	6
2.4.4 돌발상황정보(104)	7
2.4.5 도로상태정보(105)	9
2.4.6 기상정보(106)	10
2.4.7 도로관리정보(107)	11
2.4.8 프로브정보(108)	13
2.4.9 차량검지정보(109)	14
2.5 기본교통정보교환을 위한 통신프로토콜	15
2.5.1 구축 시 고려사항	15
2.5.2 데이터패킷 구조	16
2.5.3 정보교환절차	18
2.6 정보센터 Address(ID)부여	25

2.6.1 정보센터ID .....	25
2.6.2 정보센터ID 부여방법 .....	25
2.7 기술기준 준수여부 확인을 위한 검사방법 .....	26
2.7.1 검사 항목 도출 .....	26
2.7.2 검사 방법 .....	27
3. 기본교통정보교환 기술기준 II .....	29
3.1 개요 .....	29
3.2 기술기준 구성 및 적용범위 .....	29
3.3 정보의 구성 .....	31
3.4 교통정보세항목 구현방법 .....	31
3.4.1 교통소통정보 .....	31
3.4.2 교통통제정보 .....	34
3.4.3 돌발상황정보 .....	37
3.4.4 도로상태정보 .....	39
3.5 패킷구조 및 교환방법 .....	41
3.5.1 패킷구조 및 입력방법 .....	41
3.5.2 데이터부 구성 .....	43
3.5.3 데이터교환절차 .....	45
4. 기본교통정보교환 기술기준 IV .....	46
4.1 개요 .....	46
4.2 기술기준 구성 및 적용범위 .....	46
4.3 정보의 구성 .....	46
4.4 정보 연계체계 .....	47
4.5 교통정보세항목 구현방법 .....	49
4.5.1 교통소통정보 .....	49

4.5.2 교통통제정보 .....	51
4.5.3 돌발상황정보 .....	54
4.5.4 도로상태정보 .....	57
4.5.5 기상정보 .....	58
4.5.6 <u>프로브정보</u> .....	60
4.6 통신절차 .....	62
<b>5. 대중교통(버스) 정보교환 기술기준 .....</b>	<b>63</b>
5.1 개요 .....	63
5.2 기술기준 구성 및 적용범위 .....	63
5.3 정보의 구성 .....	64
5.4 정보연계 .....	66
5.5 교통정보세항목 구현방법 .....	68
5.5.1 버스위치정보(201) .....	68
5.5.2 도착예정정보(202) .....	72
5.5.3 운행계획정보(203) .....	75
5.5.4 운행지시정보(204) .....	78
5.5.5 운행관리정보(205) .....	80
5.5.6 긴급상황정보(206) .....	84
<b>6. DSRC를 이용한 ETCS의 정보교환 기술기준 .....</b>	<b>88</b>
6.1 개요 .....	88
6.2 기술기준의 목적 및 범위 .....	88
6.2.1 기술기준의 목적 .....	88
6.2.2 기술기준의 범위 .....	88
6.3 기술기준의 구성 .....	89
6.3.1 명령어 .....	89

6.3.2 요금처리 절차(ETC-Debit Transaction flow) .....	90
6.3.3 정보형식(Message format) .....	90
6.3.4 다양한 카드의 처리 방안 .....	90
<b>6.4 응용인터페이스 .....</b>	<b>91</b>
6.4.1 DSRC와 ETCS 응용프로세스의 관계 .....	91
6.4.2 서비스 프리미티브 .....	92
<b>6.5 데이터를 이용한 요금징수절차 .....</b>	<b>94</b>
6.5.1 ETCS 요금징수절차의 세부 기능 .....	97
6.5.2 INITIALIZATION.Request .....	98
6.5.3 INITIALIZATION.Response .....	99
6.5.4 Get.Request .....	101
6.5.5 Get.Response .....	102
6.5.6 Set.Request .....	103
6.5.7 Set.Response .....	104
6.5.8 EVENT-REPORT.Request .....	104
6.5.9 EVENT-REPORT.Response .....	105
6.5.10 ACTION.Request .....	106
6.5.11 ACTION.Response .....	106
6.5.12 리턴값 .....	107
6.5.13 Context Mark(EID) 구성 .....	108
<b>제2장 기술기준 적용 이해 .....</b>	<b>109</b>
<b>1. ASN.1 범용툴을 이용한 기술기준 적용 방법 .....</b>	<b>109</b>
1.1 ASN.1 범용툴 개요 .....	109
1.1.1 개발 배경 .....	109

1.1.2 ASN.1 범용툴 기능 .....	109
1.1.3 ASN.1 범용툴 시스템 아키텍처 .....	110
<b>1.2 ASN.1 범용툴 설치 .....</b>	<b>114</b>
1.2.1 범용툴 획득 .....	114
1.2.2 범용툴 설치과정 .....	115
1.2.3 범용툴 라이센스 발급 .....	116
<b>1.3 ASN.1 범용툴 기본 사용방법 .....</b>	<b>117</b>
1.3.1 ASN.1 컴파일 생성 .....	117
1.3.2 Type Descriptor .....	120
1.3.3 Support Type 및 보조 함수 .....	129
1.3.4 범용툴을 이용한 컴파일 및 예제 .....	145
<b>1.4 BIS 개발 시 ASN.1 범용툴 활용 및 사례 .....</b>	<b>148</b>
1.4.1 ASN.1의 이해 .....	148
1.4.2 어플리케이션 개발 활용 .....	149
1.4.3 ASN.1 사용시 주의 할 점 .....	157
<b>2. 표준적용 확인 방법 .....</b>	<b>159</b>
2.1 개요 .....	159
2.1.1 ITS 표준준수 여부 확인의 개념 .....	159
2.1.2 ITS 표준준수 여부 확인의 필요성 .....	160
2.2 ITS 표준준수 여부 확인방법 .....	162
2.2.1 ITS 표준준수 여부 확인을 위한 검사 .....	162
2.2.2 ITS 표준준수 여부 확인 대행 .....	162
<b>참고문헌 .....</b>	<b>170</b>

## - 표 목차 -

<표 1> 기술기준 목록 .....	1
<표 2> 기본교통정보교환 기술기준의 기본교통정보 항목 .....	3
<표 3> Header부 구성요소 .....	16
<표 4> 인증정보(Authentication Information) .....	17
<표 5> 데이터부(PDU)의 유형 .....	17
<표 6> 정보센터 ID 부여 (예) .....	25
<표 7> 기본교통정보교환 기술기준 II의 정보항목 .....	31
<표 8> 기본교통정보교환 기술기준 II의 통제유형의 주요 코드값 .....	35
<표 9> 기본교통정보교환 기술기준 II의 교통통제 시간정보항목 .....	37
<표 10> 기본교통정보교환 기술기준 II의 돌발상황발생정보 시간항목 값 입력(예) .....	38
<표 11> 도로 및 차로차단 경우의 수에 따른 선택항목 입력(예) .....	41
<표 12> 통신망의 특성에 따른 데이터 교환절차 .....	45
<표 13> 기본교통정보교환 기술기준 IV의 정보항목 구성 .....	47
<표 14> 기본교통정보교환 기술기준 IV의 교통통제정보 .....	52
<표 15> 기본교통정보교환 기술기준 IV의 교통통제 시간정보항목 .....	53
<표 16> 기본교통정보교환 기술기준 IV의 돌발상황발생정보 시간항목값 입력(예) .....	55
<표 17> 기본교통정보교환 기술기준 IV의 도로 및 차로차단 경우의 수 .....	58
<표 18> 기상예측유무 및 기상측정상태 코드 .....	59
<표 19> 차량분류 코드 .....	60
<표 20> 대중교통(버스)정보교환 기술기준의 적용범위 .....	64
<표 21> 대중교통(버스)정보교환 기술기준 정보항목 .....	65
<표 22> 대중교통(버스)정보교환 기술기준 정보항목 분석 .....	66
<표 23> 버스위치정보/ 이벤트정보 정의내용 .....	71
<표 24> 버스위치정보/ 정주기정보 정의내용 .....	71
<표 25> 도착예정정보 정의내용 .....	74
<표 26> 운행계획정보 정의내용 .....	77
<표 27> 운행지시정보 정의내용 .....	79

〈표 28〉 운행관리정보 정의내용 .....	82
〈표 29〉 운행관리정보의 상태코드별 판단방법론 .....	83
〈표 30〉 긴급상황정보 정의내용 .....	86
〈표 31〉 Context Mark(EID) 구성 .....	108
〈표 32〉 ASN.1 범용툴 기능 .....	110
〈표 33〉 범용툴 메뉴 기능 .....	113
〈표 34〉 사업별 ASN.1 범용툴 배포현황(2015년 8월 기준) .....	115
〈표 35〉 Support Library 생성 API .....	120
〈표 36〉 대중교통(버스)정보교환 기술기준의 교환정보 .....	151
〈표 37〉 ITS 표준준수여부 확인 대상 기술기준 및 검사방법 표준 .....	163
〈표 38〉 ITS 표준준수여부 확인 검사 신청 시 첨부서류 .....	165
〈표 39〉 정보제공시험항목 .....	168
〈표 40〉 정보요청시험항목 .....	168

## - 그림 목차 -

<그림 1> 교통정보 속성에 따른 연계체계	3
<그림 2> 검지기 식별번호(예)	14
<그림 3> 교통정보 교환 절차	18
<그림 4> 세션설정 절차	19
<그림 5> 세션유지 절차	20
<그림 6> 세션종료 절차	21
<그림 7> 정보요청절차	22
<그림 8> 정보제공절차	23
<그림 9> 정보센터 ID 부여 방법	25
<그림 10> 기본교통정보교환 기술기준 표준준수검사 항목	26
<그림 11> ITS 기술기준 준수 여부 검사방법	27
<그림 12> ITS 표준준수 여부 확인 검사방법	28
<그림 13> 기본교통정보교환 기술기준 II 적용범위	30
<그림 14> 기본교통정보교환 기술기준 II의 교통통제발생 예시	36
<그림 15> 기본교통정보교환 기술기준 II의 기본패킷구조	41
<그림 16> 기본교통정보교환 기술기준 IV의 상향식 교통정보 연계도	48
<그림 17> 기본교통정보교환 기술기준 IV의 하향식 교통정보 연계도	48
<그림 18> 교통통제발생 예시	53
<그림 19> 기본교통정보교환 기술기준 IV의 돌발상황 생성주기 관리 (예)	56
<그림 20> 센터-노변장치 간 데이터교환절차	62
<그림 21> 노변장치-차량장치 간 데이터교환절차	62
<그림 22> 대중교통(버스)정보교환 기술기준 정보항목 특성	65
<그림 23> 센터 간 정보연계에 따른 정보흐름	67
<그림 24> 기술기준의 범위	89
<그림 25> DSRC와 ETCS 응용프로세스의 관계	91
<그림 26> 초기서비스 교환의 논리적 절차	94
<그림 27> ETCS 요금징수절차 기본 구성	94

<그림 28> ASN.1 범용툴 시스템 아키텍처 구성도	110
<그림 29> Support Library Layer 구조	111
<그림 30> Repository 구조도	111
<그림 31> ASN.1 GUI 화면 구성	113
<그림 32> 국가 ITS 데이터등록소를 활용한 범용툴 배포 방법	114
<그림 33> 컴파일러의 일반 구조	117
<그림 34> $A := B + 3;$ 의 추상 구문 트리	118
<그림 35> ASN.1 컴파일 과정	119
<그림 36> ITS 표준준수 여부 확인의 개념	159
<그림 37> ITS 표준준수여부 확인 신청 및 진행 절차	164
<그림 38> 시험환경구성	166
<그림 39> 시험항목 분류	166
<그림 40> 기본시험과 통합시험 개념도	167

# 제1장

## 기술기준 이해

### 1. 개요

- 국토교통부는 교통정보 수집, 관리, 제공의 호환성과 연계성을 제고하여 지능형교통체계의 활용성을 높이고 더욱 다양하고 종합적인 ITS 서비스를 제공하기 위해 국가통합교통체계효율화법 제82조에 따라 ITS 표준을 제정·고시하고 있으며, 강제성이 없는 국가표준과의 차별성을 두기 위해 이를 '기술기준'이라 명명함
  - 국토교통부는 현재까지 총 5개의 기술기준을 제정·고시함(2016년 9월 기준)
- 기술기준은 교환할 정보에 대한 구성 및 정의와 이에 대한 ASN.1 표기방법을 정하고 있으며, 정보교환을 위한 프로토콜을 함께 명시하고 있음

<표 1> 기술기준 목록

고시번호	기술기준 명	적용 대상	적용범위
제2016-206호 (개정)	기본교통정보교환 기술기준	‣ 도로교통시설을 관리하는 센터	센터-센터 간 상호 교통정보교환
제2016-186호 (개정)	대중교통(버스)정보교환 기술기준	‣ 공공센터 및 연계시스템 ‣ 노선버스	공공시스템과 연계하는 시스템 간의 교통정보교환
제2016-207호 (개정)	기본교통정보교환 기술기준 II	‣ 도로교통시설을 관리하는 센터 ‣ 공중단말장치 : KIOSK 등 ‣ ITS사업의 일환으로 개발·구축 한 PDA 등 개인단말장치 ‣ 대국민서비스를 위해 교통정보를 제공하는 장치(VMS 제외)	정보수집 및 제공을 위한 센터-단말장치 간 교통정보교환 (공중단말장치 포함)
제2013-251호 (개정)	근거리전용통신(DSRC)를 이용한 자동요금징수시스템 (ETCS)의 정보교환 기술기준(노변-단말 간)	‣ DSRC를 기반으로 하는 노변 통신장치 및 차량단말기 (요금을 주고받는 경우 적용)	DSRC 기반의 자동요금징수를 위해 노변통신장치-차량단말기 간 교통정보교환
제2016-208호 (개정)	기본교통정보 교환 기술기준 IV - 무선통신 기술을 이용한 교통정보 수집·제공 기술표준	‣ 교통정보를 활용하거나 이를 이용한 장비 (차량단말장치/ 노변통신장치) ‣ 도로교통시설을 관리하는 센터	교통정보 수집·제공을 위한 센터-노변통신장치-차량 장치 간 교통정보교환

## 2. 기본교통정보교환 기술기준 |

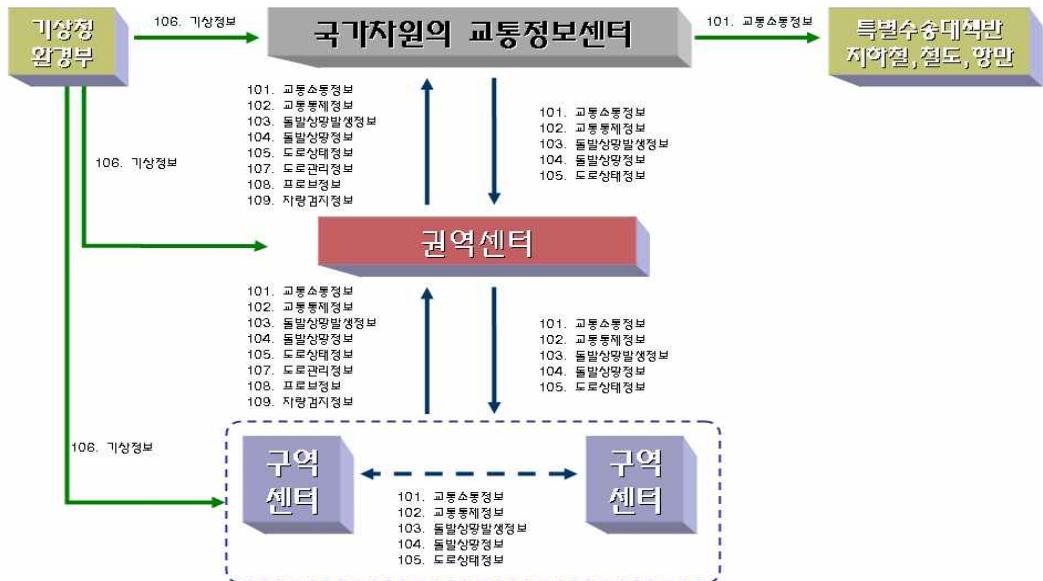
### 2.1 개요

- IT기술의 발전 및 확산에 따라 한국도로공사, 경찰청, 자치단체 등 다양한 운영 주체들이 개별적·독자적으로 ITS 구축을 추진하였으며, 교통정보의 효용가치를 높이고 시스템의 경제성, 효율성을 확보하기 위해 광역화된 생활권역 수준으로 연계·확대가 필요하게 되었음
- 이에 ITS 시스템의 상호운용성 및 호환성을 확보하기 위해 건설교통부(現 국토 교통부)는 전국적으로 구축되어 있는 ITS 시스템간의 교통정보를 상호교환하기 위하여 표준화된 데이터 표현방식을 정의한 「기본교통정보교환 기술기준」을 제정·고시함
  - 제정 2004년 12월 31일 건설교통부 고시 제2004-513호
  - 최종개정 2016년 4월 15일 국토교통부 고시 제2016-206호 (일부개정)

### 2.2 기술기준 구성 및 적용범위

- 고속국도·국도·지방도·시/군도 등 교통시설에 대한 지능형교통체계 구축·운영 시 필요한 기본적인 교통정보 교환을 위한 기술기준을 정의함으로써 교통정보 관리·제어·분석·제공의 호환성과 연계성 제고를 목적으로 함
- 본 기술기준은 정보연계체계, 기본교통정보 표준정보형식, 통신프로토콜, 센터 ID 부여방식에 대해 규정하고 있음
  - 정보연계체계 : 센터의 특성 및 위계에 따라 송·수신되는 기본교통정보 항목과 연계체계에 대해 정의
  - 기본교통정보 표준정보형식 : ITS 구축·운영 시 필요한 기본적인 교통정보를 9 개의 항목으로 추출하여 센터 간 상호 교환이 가능하도록 표준화된 정보형식으로 정의
  - 통신프로토콜 : 센터 간 기본교통정보를 교환하기 위한 규칙으로 패킷구조와 통신절차에 대해 정의
  - 센터 ID 부여방식 : 각 센터를 식별하기 위한 ID 부여방식에 대해 정의

- 기술기준의 적용범위는 도로교통시설을 관리하는 각 단위구간 또는 단위 관리청 별 구역센터, 광역적으로 통합하는 권역센터, 국가 통합센터 간 정보교환에 한함



<그림 1> 교통정보 속성에 따른 연계체계

## 2.3 정보의 구성

- 본 기술기준을 구성하는 9개의 기본교통정보는 다음 표와 같이 정의함

<표 2> 기본교통정보교환 기술기준의 기본교통정보 항목

ID	정보명	아키텍처 정보명	정보주기	정보세부항목
101	교통소통 정보	교통정보, 고속도로교통정보 도시부간선도로국도/지방도 교통정보	상시교환	속도, 교통량, 밀도, 통행시간, 대기길이, 점유율
102	교통통제 정보	교통통제정보	이벤트 발생시	위치, 통제 유형, 대상, 시간
103	돌발상황 발생정보	돌발상황정보, 돌발상황발생정보, 구조요청정보, 접보	유고상황 발생시	위치, 시각, 사상자수, 피해정도
104	돌발상황 정보	돌발상황정보, 돌발상황정보완정보 돌발상황종료정보	유고상황 발생시	관리기관, 상황유형, 대상유형, 조치상태, 간신상태
105	도로상태 정보	도로정보	요청시	노면상태, 이용가능여부, 강우/강설수위, 표면온도
106	기상정보	기상정보	요청시	기온, 날씨, 확률, 가시거리, 풍속, 풍향, 습도, 기압, 일출·일몰시간
107	도로관리 정보	도로정보	정적정보	위치, 관할구역, 도로유형, 도로명, 길이, 포장유형, 운영조건, 중앙분리형태, 자선수, 노견폭
108	프로브 정보	프로브정보, 위치정보	상시교환	차량종류, 검지시간, 통행시간, 검지위치
109	차량감지 정보	차량감지정보	상시교환	검지위치, 속도, 교통량, 점유율, 대기길이

## 2.4 교통정보세항목 구현방법

### 2.4.1 교통소통정보(101)

#### 101 교통소통정보

본 정보항목은 도로의 교통소통정보를 규약함

정보명	교통소통정보
아기부처상 정보명	교통정보, 고속도로교통정보, 도시부 간선도로교통정보, 국도/지방도 교통정보
정보세부항목	속도, 교통량, 밀도, 통행시간, 대기길이, 점유율
설명	도로의 교통소통상황을 나타내는 정보

CURLINKST DEFINITIONS AUTOMATIC TAGS::=BEGIN		정보의 이름/시작
CurrentLinkState::=SEQUENCE		
{		
link-LinkidNumber	UTF8String (SIZE(1..40)),	링크식별번호
link-SpeedRate	INTEGER (0..300),	링크속도
link-VolumeRate	INTEGER (0..100000) OPTIONAL,	링크교통량
link-DensityRate	INTEGER (0..2000) OPTIONAL,	링크밀도
link-TravelTimeQuantity	INTEGER (0..10800) OPTIONAL,	링크통행시간
link-DelayQuantity	INTEGER (0..12000) OPTIONAL,	링크지체
tfdt-VehicleQueueLengthQuantity	INTEGER (0..500000) OPTIONAL,	차량대기행렬길이
tfdt-OccupancyPercent	INTEGER (0..100)	검지기 점유율
}		
END		

가공정보  
검지 정보

명칭	설명	단위
링크식별번호	네트워크내 링크에 대한 고유식별번호	번호
링크속도	주어진 시간동안에 링크를 주행한 차량의 평균 운행속도	km/시
링크교통량	주어진 시간동안에 링크를 통과한 차량	대/시
링크밀도	링크의 단위 킬로미터당 차량수	대/시
링크통행시간	주어진 시간동안에 링크를 주행한 차량의 평균통행시간	초
링크지체	특정링크를 통행하는 차량의 지체시간	초
차량대기행렬길이	검지기에 의해 측정된 차량들의 평균대기행렬 길이	m
검지기점유율	검지기 점유 백분율 (검지된 시간/총 단위시간×100)	%

- 링크식별번호는 표준 노드·링크체계의 ID를 따름
  - 「지능형교통체계 표준 노드·링크 구축기준(국토교통부 고시 제2015-75, 일부개정)」 및 「지능형교통체계 표준 노드·링크 구축·관리 지침(국토교통부 고시 제2015-755호, 전부개정)」 참조
- 링크의 속도, 교통량, 밀도, 통행시간, 지체정보는 가공된 정보로써, 108.프로브정보와 109.차량검지정보로부터 수집된 교통정보로부터 산출된 정보이며, 차량대기행렬길이와 검지기점유율은 109.차량검지정보로부터 얻어진 원천정보를 사용함

## 2.4.2 교통통제정보(102)

- 미리 계획된 도로 및 주변 시설의 공사·집회 및 특수상황에 대처하기 위하여 사전에 계획하는 교통통제정보를 말함
- 교통통제정보는 어떠한 원인에 의해 도로의 사용이 제한되는 경우, 이에 대한 정보를 시작시간과 종료시간을 함께 제공함으로써 도로를 이용하는 이용자들에게 경로선택의 기회를 제공할 수 있음

### 102 교통통제정보

본 정보항목은 도로의 이벤트에 따른 교통통제정보의 정보세부항목을 규약함

정보명	교통통제정보
아키텍처상 정보명	교통통제정보, 도로정보
정보세부항목	위치, 통제유형, 대상, 시간
설명	미리 계획된 도로의 공사, 행사 등 교통의 통제상황을 나타내는 이벤트 정보

```

EVENTIDENTITY DEFINITIONS AUTOMATIC TAGS ::= BEGIN
EventIdentity ::= SEQUENCE
{
node-NodeIdNumber          UTF8String (SIZE(1..40)),
link-LinkIdNumber           UTF8String (SIZE(1..40)),
evnt-IdentificationNumber   INTEGER (0..4294967295) OPTIONAL,
evnt-LocationJurisdictionNameText UTF8String (SIZE(1..256)),
evnt-LocationLinkIdNumber   UTF8String (SIZE(1..40)),
evnt-LocationNodeIdNumber   UTF8String (SIZE(1..40)),
evnt-LocationRoadwayNameText UTF8String (SIZE(1..256)),
evnt-DescriptionTypeEventCode ENUMERATED          OPTIONAL,
evnt-DescriptionTypeSpecialEventCode ENUMERATED          OPTIONAL,
levnt-DescriptionPlannedRoadwayClosureCode ENUMERATED          OPTIONAL,
evnt-LanesBlockedOrClosedNumber INTEGER (0..255),
evnt-LanesShouldersBlockedCode BIT STRING,
evnt-TimelinestartGeneralizedTime VisibleString,
evnt-TimelineEndGeneralizedTime  VisibleString,
evnt-TimelineDurationTime    INTEGER (0..4294967295)
}
END

```

정보의 이름/시작

노드식별번호

링크식별번호

이벤트식별번호

이벤트위치관할구역명

이벤트위치 링크

이벤트위치 노드

위치도로명칭

이벤트유형 구분

없음(0), 기타(1), 공사(2),  
도로폐쇄(3), 특별행사(4)  
**특별이벤트유형설명**

없음(0), 기타(1), 퍼레이드(2),  
스포츠이벤트(3), 시위및집회(4),  
축제(5), 예술공연(6),  
수행행렬(7)

계획된도로차단유형설명

없음(0), 기타(1), 장기공사(2),  
단기공사(3), 유지보수(4),  
긴급유지보수(5)

차단차로수

차선길어깨차단코드

없음(0), 기타(1), 우측길어깨(2),  
좌측길어깨(3), 양쪽길어깨(4)

이벤트개시시각

이벤트종료시각

이벤트지속표준시

위  
치  
정  
보

이  
벤  
트  
정  
보

규  
제  
정  
보

시  
간  
정  
보

명칭	설명	단위
노드식별번호	네트워크내 노드에 대한 고유식별번호	번호
링크식별번호	네트워크내 링크에 대한 고유식별번호	번호
이벤트식별번호	도로 이벤트에 대해 교통관제센터에서 사용되는 고유식별번호	번호
이벤트위치 관할구역명	도로 이벤트가 발생한 도로에 대해 책임이 있는 행정 관할기관 명	문자
이벤트 위치링크	도로 이벤트의 발생 위치와 관련된 링크 번호	번호
이벤트 위치노드	도로 이벤트와 관련된 노드번호	번호
위치도로명칭	도로 이벤트가 발생한 도로의 명칭	문자
이벤트유형구분	각종 도로 이벤트의 유형을 구분할 수 있는 코드	-
특별이벤트 유형설명	도로 이벤트 중 특별한 도로 이벤트의 유형을 표현하기 위한 코드	-
계획된 도로차단유형설명	도로 이벤트 중 계획된 도로차단의 유형을 표현하기 위한 코드	-
차단차로수	도로 이벤트로 인해 차단/통제된 차로수	수
차선길어깨 차단코드	도로 이벤트로 인해 차단된 길어깨를 나타내는 코드	-
이벤트개시시각	특별한 도로 이벤트의 실제적인 개시 시각	날짜, 시각
이벤트종료시각	도로 이벤트의 실제적인 종료 시간	날짜, 시각
이벤트 지속표준시	도로 이벤트의 실제적인 진행 시간	분

### 2.4.3 돌발상황발생정보(103)

#### 103 돌발상황발생정보

본 정보항목은 돌발상황의 위치와 내용에 대해 규약함

정보명	돌발상황발생정보	정보의 이름/시작
이카텍처상 정보명	돌발상황발생정보, 돌발상황정보, 구조요청, 접보	
정보세부항목	위치, 시각, 사상자수, 피해정도	
설명	돌발상황발생시 센터로 제보되거나 구조요청을 통해 접수된 돌발상황의 위치 및 발생시기정보	

INCIDENTIDENTITY DEFINITIONS AUTOMATIC TAGS ::= BEGIN IncidentIdentity ::= SEQUENCE { node-NodeIdNumber UTF8String (SIZE(1..40)), link-LinkIdNumber UTF8String (SIZE(1..40)), evnt-LocationLinearDistanceOffsetBeginNumber INTEGER (0..999) OPTIONAL, evnt-LocationLinearDistanceOffsetEndNumber INTEGER (0..999) OPTIONAL, evnt-UpdateGeneralizedTime VisibleString, evnt-IncidentHumanFatalitiesCountQuantity INTEGER (0..255), evnt-IncidentHumanInjuriesCountQuantity INTEGER (0..255), evnt-IncidentPropertyDamageCode BIT STRING,  evnt-PropertyDamageIncidentOther UTF8String (SIZE(1..256)) OPTIONAL } END	정보의 이름/시작
---	-----------

명칭	설명	단위
노드식별번호	네트워크 내 노드에 대한 고유식별번호	번호
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
위치선형거리옵셋 시점	도로 및 경로 또는 링크 상의 거리로써 측정된 도로 이벤트 발생구간 시점의 위치	0.1m
위치선형거리옵셋 종점	도로 및 경로 또는 링크 상의 거리로써 측정된 도로 이벤트 발생구간 종점의 위치	0.1m
돌발상황갱신시각	특별한 도로 이벤트에 대하여 최근 변경시 갱신되어진 시각	시각
돌발상황사망자수	보고된 돌발상황에 의해 사망한 사람수	명
돌발상황부상자수	보고된 돌발상황에 의해 부상한 사람수	명
돌발상황재산피해정도	돌발상황으로 인한 재산상 피해의 유형을 나타내는 코드	-
돌발상황재산피해정도 부연설명	피해상황 부연설명	문자

- 돌발상황발생정보는 돌발상황의 발생에 따른 신속한 정보수집·전달을 위한 것으로 돌발상황발생 자체에 대한 신속한 정보전달에 의미가 있기 때문에 많은 정보를 포함하지 않고 간략한 정보만 포함함
- 돌발상황발생정보는 돌발상황으로 인한 물적·인적 피해정보만을 우선적으로 제공하는 것이므로 추후 104.돌발상황정보와 대응되어 돌발상황의 원인 및 유형 등이 종합적으로 관리되기 위해서 각 센터별로 돌발상황별 개별 ID를 부여하여 관리하는 것이 바람직 함

#### 2.4.4 돌발상황정보(104)

- 돌발상황정보는 도로상 또는 인접지역에서 돌발적으로 발생하여 교통류 흐름에 장애를 주거나 또는 장애요인의 가능성 있는 상황에 대한 정보임
- 돌발상황정보는 돌발상황의 유형, 종료시간 뿐 아니라 처리과정에 대한 정보 등 돌발상황에 대한 세부적인 정보를 포함하고 있으며 103.돌발상황발생정보의 부가적 정보로 사용될 수 있음

## 104 돌발상황정보

본 정보항목은 돌발상황의 처리사항에 대해 규약함

정보명	돌발상황정보
아키텍처상 정보명	돌발상황정보, 돌발상황발생정보, 돌발상황처리요청
정보세부항목	관리기관, 상황유형, 대상유형, 조치상태, 간선상태
설명	돌발상황의 처리상태를 각 센터 및 수집제공단에 제공하는 정보

```

INCIDENTCONDITIONS DEFINITIONS AUTOMATIC TAGS ::= BEGIN
IncidentConditions ::= SEQUENCE
{
node-NodeldNumber           UTF8String    (SIZE(1..40)),
link-LinkIdNumber            UTF8String    (SIZE(1..40)),
orgn-ContactOrganizationNameText UTF8String    (SIZE(1..128)),
evnt-DescriptionTypeIncidentCode ENUMERATED,
                                         {
                                         evnt-DescriptionTypeIncidentOther      UTF8String    (SIZE(1..256)) OPTIONAL,
                                         evnt-IncidentVehiclesInvolvedCode     BIT STRING,
                                         evnt-IncidentVehiclesInvolvedOther     UTF8String    OPTIONAL,
                                         evnt-IncidentStatusCode                ENUMERATED,
                                         evnt-IncidentStatusOther              UTF8String    (SIZE(1..256)) OPTIONAL,
                                         evnt-UpdateTypeCode                  ENUMERATED,
                                         evnt-UpdateTypeOther                 UTF8String    (SIZE(1..256)) OPTIONAL
                                         }
                                         END

```

정보의 이름/시작

노드식별번호

링크식별번호

연락기관명칭

돌발상황의 유형설명

추가정보필요없음(0),  
추가정보필요(1), 차량사고(2),  
기상관련사고(3),  
기후고장등으로인한 차량의 정지(4),  
차량화재(5), 장애물(6),  
위험물질방출(7), 지진(8),  
산사태(9), 홍수(10), 태풍(11),  
예고되지않은시위/집회(12),  
차량의 급격한증가(13)

돌발상황유형의 부연설명

돌발상황대상차량

추가정보필요없음(0),  
추가정보필요(1),  
자전거/오토바이(2), 승용차(3),  
버스(4), 트럭(5), 트레일러(6)

돌발상황대상차량의 부연설명

돌발상황상태

돌발상황상태의 부연설명

간선유형코드

간선유형의 부연설명

위치정보

돌발상황정보

명칭	설명	단위
노드식별번호	네트워크 내 노드에 대한 고유식별번호	번호
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
연락기관명칭	지역 내의 특정 기관의 명칭. 상호 관련된 기관이 있을 경우 최상위 기관을 명시	문자
돌발상황의 유형설명	돌발상황에 대한 유형을 표현하기 위한 코드	-
돌발상황유형의 부연설명	돌발상황의 유형에 대한 부연설명	문자
돌발상황 대상차량	돌발상황과 관련된 차량의 유형을 나타내는 코드	-
돌발상황 대상차량의 부연설명	돌발상황 대상차량의 부연설명	문자
돌발상황상태	돌발상황 상태를 나타내는 코드	-
돌발상황상태의 부연설명	돌발상황 상태의 부연설명	문자
간선유형코드	특별한 도로 이벤트에 대해 간선되어진 유형을 나타내는 코드	-
간선유형의 부연설명	간선유형의 부연설명	문자

## 2.4.5 도로상태정보(105)

- 도로노면상태에 관련된 정보를 제공함으로써 도로이용자에게 안전운행을 유도할 뿐 아니라 신속한 안전처리를 통해 정상적 운행을 가능하게 함
- 돌발상황정보와 같이 2개 이상의 노드 또는 링크에 대한 도로상태정보를 전송하기 위해서는 구분자를 사용하여 노드와 링크값을 입력 가능

### 105 도로상태정보

본 정보항목은 도로노면의 현재 상태 정보를 규약함

정보명	도로상태정보		
아카티처상 정보명	도로정보		
정보세부항목	노면상태, 이용가능여부, 강우/설량수위, 표면온도		
설명	기상조건 및 돌발상황에 따른 도로의 상태정보		

RDWSFST DEFINITIONS AUTOMATIC TAGS ::= BEGIN			정보의 이름/시작
RoadwaySurfaceStatus ::= SEQUENCE			
{			
node-NodeIdNumber	UTF8String (SIZE(1..40)),	노드식별번호	도로 노 면 통 제 정 보
link-LinkIdNumber	UTF8String (SIZE(1..40)),	링크식별번호	
link-SurfaceConditionCode	BIT STRING,	도로노면상태	
		추가정보필요없음(0), 추가정보필요(1), 결정사항없음(2), 노면습윤상태(3), 결빙(4), 적설(5), 낙석(6), 기름유출(7), 잔해(8), 포장상태나쁨(9), 전력선단선으로 인한 도로점거/점유(10), 적재물낙하(11), 화학물질 유출(12) 침수(13)	
Link-LandBlockedOrClosedCode	BIT STRING,	도로차단여부	
evsn-DataSurfacePrecipitationDepth	INTEGER (0..3000) OPTIONAL,	추가정보필요없음(0), 추가정보필요(1), 결정사항없음(2), 차단안됨(미용가능)(3), 차단됨(미용불가)(4)	
evsn-DataSurfaceTemperatureQuantity	INTEGER (0..300) OPTIONAL	강우/설량수위	
}			
END		도로표면온도	

명칭	설명	단위
노드식별번호	네트워크 내 노드에 대한 고유식별번호	번호
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
도로노면상태	링크의 현재 도로노면 상태 (코드)	-
도로차단여부	도로차단여부 (코드)	-
강우/설량 수위	환경감지기에서 도로 노면의 강우량 수위를 나타내는 숫자적인 색인	mm
도로표면온도	환경감지기에서 섭씨로 측정된 도로의 표면 온도	°C

## 2.4.6 기상정보(106)

- 현재의 기상정보와 예측된 기상정보 모두 제공이 가능하며 정보 요청 시에만 기상청, 환경부로부터 제공받을 수 있음
  - 기상예측유무는 필수로 제공하여야 하며, 이외의 메시지는 선택적으로 제공

### 106 기상정보

본 정보항목은 해당구간의 기상상황 정보를 규약함

정보명	기상정보			
아키텍처상 정보명	기상정보			
정보세부항목	기온, 날씨, 확률, 가시거리, 풍속, 풍향, 습도, 기압, 일출, 일몰시간			
설명	일반적인 기상상황을 나타내는 기상정보			

WETHININFO DEFINITIONS AUTOMATIC TAGS ::= BEGIN		정보의 이름/시작	
WeatherIdentity ::= SEQUENCE {		기상예측유무	
wthr-ForecastOrActualCode	ENUMERATED,	현황기상정보(0), 예측기상정보(1)	
wthr-TemperatureQuantity	INTEGER (-128..127) OPTIONAL,	기온	
wthr-HighTemperatureQuantity	INTEGER (-128..127) OPTIONAL,	최고기온	
wthr-LowTemperatureQuantity	INTEGER (-128..127) OPTIONAL,	최저기온	
wthr-ProbabilityPercent	INTEGER (0..100),	기상예보확률	
wthr-SpecialConditions	BIT STRING OPTIONAL,	기상특정상태	
-----		추가정보필요없음(0), 추가정보필요(1), 눈사태(2), 구름(3), 강추위(4), 폭염(5), 婊수(6), 안개(7), 태풍(8), 오존경보(9), 호우/대설(10), 뇌우(11), 자외선경보(12)	
wthr-SpecialConditionsOther	UTF8String (SIZE(1.. 256)) OPTIONAL,	기상특정상태의 부연설명	
wthr-VisibilityQuantity	INTEGER (0..255) OPTIONAL,	가시거리	
wthr-WindSpeedQuantity	INTEGER (0..255) OPTIONAL,	풍속	
wthr-WindDirectionCode	ENUMERATED OPTIONAL,	풍향	
wthr-HumidityQuantity	INTEGER (0..100) OPTIONAL,	습도	
wthr-PressureQuantity	INTEGER (0..100) OPTIONAL,	기압	
wthr-SunriseTimeUTCTime	VisibleString OPTIONAL,	일출시간	
wthr-SunsetTimeUTCTime	VisibleString OPTIONAL,	일몰시간	
}			
END			

명칭	설명	단위
기상예측유무	기상의 유형으로 실제와 예측으로 구분 (코드)	-
기온	온도	°C
최고기온	최고기온	°C
최저기온	최저기온	°C
기상예보확률	기상예측에 대한 확률적 평가	%
기상특정상태	날씨정보의 특정상태로서 적란문에 의해서 생성되는 비의 대기조건 또는 태풍 등과 같은 기상현상을 설명 (코드)	-

명칭	설명	단위
기상특정상태의 부연설명	날씨정보의 부연설명	문자
가시거리	가시거리로서 m로 나타냄	m
풍속	바람의 속도	m/s
풍향	바람의 방향 (코드)	-
습도	습도	%
기압	기압	mb
일출시간	일출시간	시각
일몰시간	일몰시간	시각

## 2.4.7 도로관리정보(107)

### 107 도로관리정보

본 정보항목은 도로의 관리상황 정보를 규약함

정보명	도로관리정보					
아키텍처상 정보명	도로정보					
정보세부항목	위치, 관할구역, 도로유형, 도로명, 길이, 포장유형, 운영조건, 중앙분리형태, 차선수, 노면폭					
설명	도로의 유지관리를 위한 기본정보					
<b>TRANSITGEOMETRY DEFINITIONS AUTOMATIC TAGS ::= BEGIN</b> LinkRoadwayGeometry ::= SEQUENCE { link-BeginNodeIDNumber UTF8String (SIZE(1..40)). link-BeginNodeLocation UTF8String (SIZE(1..256)). OPTIONAL. link-EndNodeIDNumber UTF8String (SIZE(1..40)). link-EndNodeLocation UTF8String (SIZE(1..256)). OPTIONAL. link-JurisdictionText UTF8String (SIZE(1..256)). link-TypeCode BIT STRING.  link-NameText UTF8String (SIZE(1..256)). link-LengthQuantity INTEGER (0..160000). link-PavementTypeCode BIT STRING.  tfmg-LinkHeightRestrictionQuantity INTEGER (1..2000) OPTIONAL. tfmg-LinkLeftShoulderWidthQuantity INTEGER (1..999) OPTIONAL. tfmg-LinkLengthRestrictionQuantity INTEGER (1..6000) OPTIONAL. tfmg-LinkMedianTypeCode ENUMERATED.  tfmg-LinkNumLanesNumber INTEGER (1..300). tfmg-LinkNumLanesOpenNumber INTEGER (1..50). tfmg-LinkRightShoulderWidthQuantity INTEGER (1..999) OPTIONAL. } END						
<div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>정보의 이름/시작</span> </div> <div style="flex: 1;"> <span>시점노드식별번호</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>시점노드위치</span> </div> <div style="flex: 1;"> <span>종점노드식별번호</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>종점노드위치</span> </div> <div style="flex: 1;"> <span>관할구역</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>링크유형</span> </div> <div style="flex: 1;"> <span>추가정보필요없음(), 추가정보필요(1), 고속도로(2), 주간선도로(3), 보조간선도로(4), 우회도로(5), 군사도로(6), 터널(7), 즉도(8)</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>링크명칭</span> </div> <div style="flex: 1;"> <span>링크길이</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>노면포장유형</span> </div> <div style="flex: 1;"> <span>추가정보필요없음(), 추가정보필요(1), 콘크리트(2), 아스팔트(3), 구스-콘크리트(4), 강철(5), 자갈(6), 진흙(7), 임시포장(8), 미끄럼방지포장(9)</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>링크높이제한</span> </div> <div style="flex: 1;"> <span>링크좌측노면폭</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>차량길이제한</span> </div> <div style="flex: 1;"> <span>중앙분리대종류</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>전디(0), 모래(1), 차광등보호대 있는콘크리트분리대(2), 도로분리(3), 분리없음(4), 중앙선(5), 좌회전있는중앙선(6), 콘크리트분리대(7), 가드레일(8)</span> </div> <div style="flex: 1;"> <span>링크차선수</span> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <span>링크운영차선수</span> </div> <div style="flex: 1;"> <span>링크우측노면폭</span> </div> </div>						

위  
치  
정  
보

도  
로  
중  
류  
/  
구  
조  
정  
보

명칭	설명	단위
시점노드식별번호	링크 시점에 해당되는 노드의 고유식별번호	번호
시점노드위치	링크 시작점의 경위도 좌표	문자
종점노드식별번호	링크 종점에 해당되는 노드의 고유식별번호	번호
종점노드위치	링크 종료점의 경위도 좌표	문자
관할구역	링크가 포함되어 있는 행정관할구역 명칭	문자
링크유형	링크 유형의 설계구분 (코드)	-
링크명칭	이용자가 명확히 파악할 수 있는 ASCII코드로 된 링크 명칭	-
링크길이	시점노드에서 종점노드까지의 링크 길이	m
노면포장유형	도로 노면포장 물질의 유형 (코드)	-
링크높이제한	링크상의 차량의 제한높이	cm
링크좌측노견폭	링크의 좌측노견	m
차량길이제한	링크상에서 허용되는 차량의 최대길이	cm
중앙분리대종류	링크를 구분하는 중앙분리대 형태 (코드)	-
링크차선수	링크의 차선수	수
링크운영차선수	링크의 현재 운영중인 차선수	수
링크우측노견폭	링크에 대한 우측 노견의 폭	cm

- 도로관리정보는 도로의 관리 속성이 변경될 때만 교환하는 정보임
- 시점노드위치, 종점노드위치는 국제표준(IEC61162)의 위치정보 표기법 중 \$GPGGA의 경위도 좌표를 입력함
  - 예를 들면, \$GPGGA, 137809.000, 3732.6627, N, 12701.3549, E, ... 중 경위도 좌표인 3732.6627, N, 12701.3549, E를 입력

## 2.4.8 프로브정보(108)

- 프로브 정보는 주기적으로 수집되는 교통소통정보를 보완·검증·확인하기 위하여 OBU 탑재차량을 투입하여 수집하는 정보를 의미함

### 108 프로브정보

본 정보항목은 교통소통연장상황을 알려주는 프로브 정보를 규약함

정보명	프로브정보		
아키텍처상 정보명	프로브정보, 위치정보		
정보세부항목	차량종류, 검지시간, 통행시간, 검지위치		
설명	프로브 차량을 이용하여 수집되는 기본교통정보		

TRAFFICPROBE DEFINITIONS AUTOMATIC TAGS ::= BEGIN	정보의 이름/시작
ProbeVehicleDetection ::= SEQUENCE	
{	
vpbd-VehicleIdNumber	UTF8String (SIZE(1..256)).
vpbd-VehicleClass	ENUMERATED.
vpbd-VehicleTimeInGeneralizedTime	VisibleString.
vpbd-VehicleTimeOutGeneralizedTime	VisibleString.
vpbd-VehicleTravelTimeTime	INTEGER (0..10800).
vpbd-VehicleAverageSpeedRate	INTEGER (0..300) OPTIONAL.
vpbd-AviReaderIdNumber	UTF8String (SIZE(1..256)) OPTIONAL.
vpbd-AviReaderLocation	UTF8String (SIZE(1..256)) OPTIONAL.
vpbd-ReferenceLocatorIdNumber	UTF8String (SIZE(1..256)) OPTIONAL.
vpbd-ReferenceLocatoion	UTF8String (SIZE(1..256)) OPTIONAL.
}	
END	

차량식별번호	프로브 정보
차량분류코드	
추가정보필요없음(0), 추가정보필요(1), 버스(2), 택시(3), 일반승용차(4), 을급차량(5), 관용차량(6)	
차량진입시간	수집정보
차량진출시간	
차량통행시간	
차량평균속도	
AVI판독기식별번호	위치정보
AVI판독기위치	
참조위치탐지기식별번호	
참조위치탐지기위치	

명칭	설명	단위
차량식별번호	개별 프로브 차량에게 배정된 고유한 태그 식별번호	번호
차량분류코드	프로브 차량 유형에 대한 차종 분류 코드	-
차량진입시간	프로브 차량이 도로 상류부에 설치된 AVI 판독기에 감지된 시간	시각
차량진출시간	프로브 차량이 도로 하류부에 설치된 AVI 판독기에 감지된 시간	시각
차량통행시간	자동차량인식기에 의해 파악된 프로브 차량의 평균 링크 통행시간	초
차량평균속도	자동차량인식기를 통해 파악한 프로브 차량의 평균운행속도	km/시
AVI판독기식별번호	자동차량인식기의 고유식별번호	번호
AVI판독기위치	자동차량인식기가 설치된 지점의 경위도 좌표	문자
참조위치탐지기식별번호	GPS 참조 위치탐지기의 고유식별번호	번호
참조위치탐지기위치	GPS 참조 위치탐지기가 설치된 지점의 경위도 좌표	문자

- 시점노드위치, 종점노드위치는 국제표준(IEC61162)의 위치정보 표기법 중 \$GPGGA의 경위도 좌표를 입력함
  - 예를 들면, \$GPGGA, 137809.000, 3732.6627, N, 12701.3549, E, ... 중 경위도 좌표인 3732.6627, N, 12701.3549, E를 입력

## 2.4.9 차량검지정보(109)

- 차량검지정보는 도로상에 설치된 검지기로부터 수집되는 정보로써 프로브 차량으로부터 수집되는 정보와 함께 가공하여 교통소통정보를 생성·제공할 수 있음

**109 차량검지정보**

본 정보항목은 구간교통상황을 모니터링하거나 분석하기 위한 정보를 규약함

정보명	차량검지정보
아이티엑처상 정보명	차량검지정보
정보세부항목	검지위치, 속도, 교통량, 점유율, 대기길이
설명	검지기를 통해 수집된 기본교통정보

```
DETECTCOLL DEFINITIONS AUTOMATIC TAGS ::= BEGIN
DetectorCollection ::= SEQUENCE
{
    tfdt-DetectorIdNumber           UTF8String (SIZE(1..256)),
    tfdt-DetectorTimeGeneralizedTime VisibleString,
    tfdt-SpeedRate                  INTEGER (0..300),
    tfdt-VolumeRate                 INTEGER (0..100)   OPTIONAL,
    tfdt-OccupancyPercent           INTEGER (0..100),
    tfdt-VehicleQueueLengthQuantity INTEGER (0..500000) OPTIONAL
}
END
```

정보의 이름/시작

검지기식별번호

차량검지시간

검지기속도

검지기교통량

검지기점유율

차량대기행렬길이

수  
집  
정  
보

명칭	설명	단위
검지기식별번호	검지기 고유식별번호	번호
검지기속도	검지기에서 올라온 차량의 평균 운행속도	km/시
검지기교통량	검지된 차량수	대/분
검지기점유율	검지기 점유 백분율 (검지된 시간/총 단위시간×100)	%
차량대기행렬길이	검지기에 의해 측정된 차량들의 평균대기행렬 길이	m

- 센터별로 관리되는 검지기 식별번호는 표준링크ID에 설치차로, 링크상의 설치위치를 기본구성으로 하고 향후 사용 가능한 확장분을 가지는 형태로 구성할 수 있음



<그림 2> 검지기 식별번호(예)

## 2.5 기본교통정보교환을 위한 통신프로토콜

- 산업표준화법에 의해 기 제정된 한국산업규격 “KS X ISO 14827-1, 2 : ITS를 위한 센터간 데이터 인터페이스” 을 기본교통정보교환을 위한 통신프로토콜로 준용함

### 2.5.1 구축 시 고려사항

- 데이터를 교환하기 전에 센터들은 다음의 사항들에 대한 합의가 이루어져야 함

#### 1) 일반

- 모든 합의사항들이 유효한 시간
- 최대 파일전송완료 응답시간, 허트비트 지속시간, 최대 응답시간 등
- 합의된 시간 이내에 합의를 종료하는 규칙
- 서버, 클라이언트의 도메인명
- 오프라인에서의 연락주소, 전화, 팩스와 이메일 등 세부사항

#### 2) 접속 및 로그인 정보

- 클라이언트 IP 주소
- 서버의 IP 주소
- 인증 받은 클라이언트 사용자 이름의 목록
- 각 클라이언트 사용자 이름에 따른 패스워드

#### 3) 프로토콜

- 다음을 포함한 저수준 계층의 프로토콜을 선택함
  - 프리젠테이션과 세션 계층 : BER을 사용하도록 규정
  - 전송과 네트워크 계층 : TCP/IP를 사용하도록 규정
  - 데이터 링크와 물리적 계층 : Ethernet, FDDI, PPP over ISDN 등
- 데이터 그램으로 정보제공이 가능한 최대 데이터 그램 크기는 576바이트를 기본 값으로 함
- 선호되는 파일전송의 프로토콜은 FTP, TFTP임

## 2.5.2 데이터패킷 구조

### 1) 패킷 구성

- 패킷은 Header부, Data부, Tail부로 구성됨

Header부					Data부	Tail부
Datex Version No.	Authentication Info	Data Packet No.	Data Packet Priority No.	Header Options	PDU	Crc

### 2) Header부

- Header부는 패킷과 송수신부의 기본정보를 전송하기 위하여 다음과 같은 정보를 포함함
  - HeaderOptions는 실제 운영상 사용할 수도 있고 사용하지 않을 수 있는 옵션 사항임

<표 3> Header부 구성요소

항목	명칭	구분	설명
Datex Version No.	datex-Version-number	ENUMERATED	Datex 버전 번호
Authentication Info	datex-Authenticationinfo-txt	OCTET STRING	각 시스템간의 연결을 위해 각 메시지에 포함하기로 동의한 인증정보
Data Packet No.	datex-DataPacket-number	INTEGER	데이터의 패킷 번호로 0부터 1씩 증가하여 부여
Data Packet Priority No.	datex-DataPacket Priority-number	INTEGER	0~10 사이의 값을 갖는 메시지 우선순위(중요도) 표시
Header Options	datex-Origin-text	UTF8String	데이터 수집 시스템의 도메인 이름
	datex-OriginAddress-location	OCTET STRING	데이터 수집 시스템의 유일한 주소
	datex-Sender-text	UTF8String	송신부 시스템의 도메인 이름
	datex-SenderAddress-location	OCTET STRING	송신부 컴퓨터의 유일한 주소
	datex-Destination-text	UTF8String	수신부 시스템의 도메인 이름
	datex-Destination Address-location	OCTET STRING	수신부 시스템의 유일한 주소
	datex-DataPacket-time	사용자정의 타입	데이터 패킷이 생성된 시간

- AuthenticationInfo는 Data부의 정보를 구분하는 것으로써 각 정보에 따른 코드 값은 다음과 같음

<표 4> 인증정보(Authentication Information)

코드	정보(PDU)명칭	설명
0x01	Initiate	초기 통신연결을 위한 개시 요청 데이터 패킷
0x02	Login	서버에 접속하기 위한 클라이언트의 로그인 데이터 패킷
0x03	FrED	서버와 클라이언트의 연결을 유지하기 위한 확인 데이터 패킷
0x04	Terminate	연결을 종료하고자 할 때, 서버에서 클라이언트에 요청하는 데이터 패킷
0x05	Logout	접속을 종료하기 위한 클라이언트의 로그아웃 데이터 패킷
0x06	Subscription	클라이언트가 서버에 정보를 요청할 경우 송신하는 데이터 패킷
0x07	TransferDone	클라이언트가 요청한 정보를 파일형태로 제공하기 위한 데이터 패킷
0x08	Accept	클라이언트의 요청에 대한 수용
0x09	Reject	클라이언트의 요청에 대한 거부
0x20~0x40	Publication	클라이언트가 요청한 정보를 제공하기 위한 데이터 패킷
0x21	CurrentLinkState	교통소통정보
0x22	EventIdentity	교통통제정보
0x23	IncidentIdentity	돌발상황발생정보
0x24	IncidentConditions	돌발상황정보
0x25	RoadwaySurfaceStatus	도로상태정보
0x26	WeatherInformation	기상정보
0x27	LinkRoadwayGeometry	도로관리정보
0x28	ProbeVehicleDetection	프로브정보
0x29	DetectorCollection	차량검지정보

### 3) Data 부

- Data 부는 데이터 저장 부분으로, 다중유형의 데이터패킷을 통해 위에서 정의된 동일한 패킷구조 내에서 모두 송신 가능한 구조를 가짐
- Data부에 포함될 수 있는 다양한 유형의 프로토콜 데이터단위는 다음과 같음

<표 5> 데이터부(PDU)의 유형

PDU 유형	설명
Initiate(개시)	서버가 방화벽 등에 의해 보호되고 있는 경우 클라이언트의 접속을 허락하기 전송하는 데이터 유닛
login(로그인)	서버에 접속하기 위해 사용자이름, 패스워드 등의 클라이언트 정보를 담는 데이터 유닛
FrED(응답데이터)	정보의 교환이 없어도 센터간의 연결된 세션을 유지하기 위한 데이터 유닛과 로그아웃요청, 파일전송완료에 대한 응답을 위한 데이터 유닛
terminate(종료)	서버가 세션을 종료하기 위하여 클라이언트에 전송하는 데이터 유닛
logout(로그아웃)	클라이언트가 세션 종료를 요청하기 위해 서버에 전송하는 데이터 유닛
subscription(서브스크립션)	기본교통정보를 요청하기 위한 데이터 유닛
publication(퍼블리케이션)	요청된 기본교통정보를 제공하기 위한 데이터 유닛
transfer-done(전송완료)	파일에 의한 기본교통정보 제공시 파일을 다 전송받았음을 서버에 알리기 위한 데이터 유닛
accept(수용)	수신된 정보에 대한 유효 확답 또는 요청을 수용하기 위한 응답 데이터 유닛
reject(거부)	수신된 정보가 유효하지 않거나 요청을 거부하기 위한 응답 데이터 유닛

#### 4) Tail 부

- Tail부는 송/수신되는 패킷의 오류를 체크하기 위한 부분으로 ISO/IEC 3309에서 규정된 CRC16 알고리즘을 적용하여 결정된 코드값을 사용함
  - \* ISO/IEC 3309:1993 : Information technology – Telecommunications and information exchange between system – High-level data link control(HDLC) procedures – frame structure (해당 표준은 2002년에 ISO/IEC 13239로 개정됨)

항목	명칭	타입	크기	설명
Crc	datex-Crc-nbr	OCTET STRING	2	Error점검 체크코드

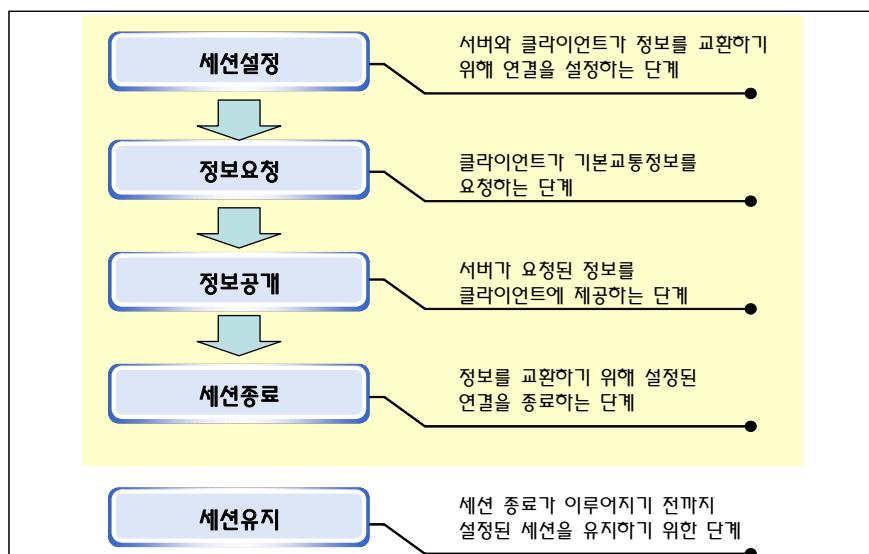
- CRC16 알고리즘에 의한 Crc값 산출식은 다음과 같으며 Crc 체크는 PDU 부분만을 체크함

$$\text{Crc값} = X^{16} + X^{15} + X^2 + 1$$

#### 2.5.3 정보교환절차

##### 1) 정보교환 절차

- 센터와 센터 간에 정보를 교환하기 위한 절차는 다음 그림과 같이 세션설정, 정보요청, 정보공개, 세션 종료, 그리고 연결된 세션을 유지하기 위한 세션 유지 단계로 구분할 수 있음



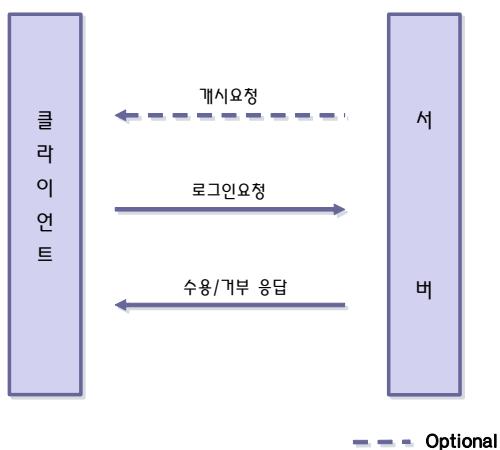
<그림 3> 교통정보 교환 절차

- 서버와 클라이언트는 상대적인 의미로써 정보를 요청하는 정보센터는 “클라이언트”, 정보를 제공하는 정보센터는 “서버”의 역할을 함

## 2) 세션설정

### (가) 세션설정 절차

- 세션 설정은 서버와 클라이언트가 교통정보를 교환하기 위한 논리적 연결로서 세션을 설정하는 절차는 다음과 같음



<그림 4> 세션설정 절차

### (나) 개시요청

- 개시요청 (initiate request)은 서버가 방화벽에 의해 보호되고 있는 경우 클라이언트의 접근을 허용하기 위해 전송하는 패킷으로 클라이언트는 로그인 요청 패킷으로 이에 응답해야 함
  - 개시요청을 위해 클라이언트는 서버의 도메인명, 클라이언트의 도메인명을 전송
- 로그인 요청패킷의 응답이 전송되지 않으면 서버는 개시요청패킷을 재전송해야 함

### (다) 로그인 요청

- 개시요청을 수신 받거나 클라이언트가 세션을 설정하려 할 경우 로그인 요청 패킷을 서버에 전송하여 세션 설정을 요청할 수 있음

- 로그인 패킷에는 도메인명, 사용자 이름, 비밀번호, 최대 허트비트지속시간, 응답제한시간, 인코딩규칙, 데이터 그램 크기, 로그인 사유에 대한 정보 포함
- 유효한 시간 내에 로그인 요청에 대한 응답이 이루어지지 않으면 클라이언트는 동일 정보의 로그인 요청 패킷을 재전송해야 함

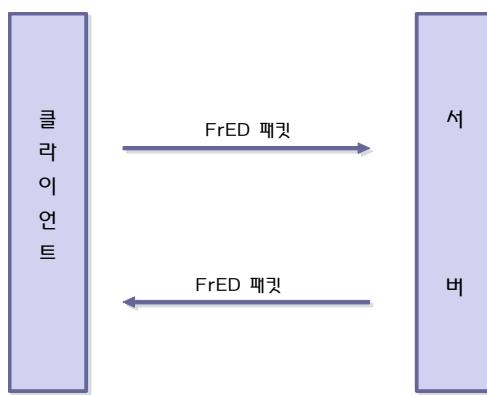
#### (라) 수용/거부 응답

- 서버는 로그인 정보가 유효하다 판단되면 로그인 요청에 대한 응답으로 수용(accept) 패킷을 전송하고 세션을 설정함
  - Data부에는 수용 타입 포함
- 서버는 로그인 정보가 유효하지 않을 경우 로그인 요청에 대한 응답으로 거부(reject) 패킷을 전송함
  - Data부에는 거부사유를 포함하여야 하며 보안상의 이유가 있을 경우에는 무응답 처리 가능

### 3) 세션유지

#### (가) 세션유지 절차

- 데이터의 교환이 이루어지지 않더라도 세션이 종료될 때까지 클라이언트는 세션을 유지하여야 하며, 이를 위해 서버와 클라이언트는 'FrED'패킷을 교환함



<그림 5> 세션유지 절차

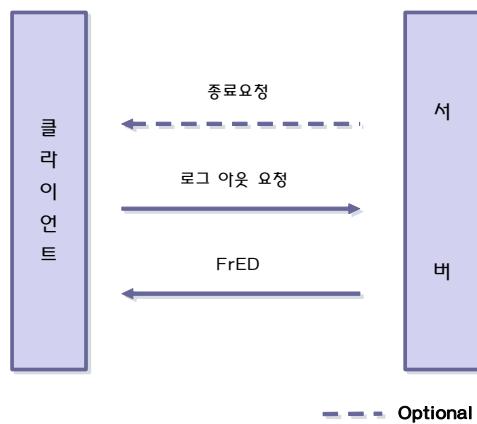
## (나) FrED

- 세션이 설정된 후 실질적으로 정보요청과 정보제공이 발생되지 않더라도 설정된 세션을 유지하기 위하여 클라이언트와 서버는 'FrED'라는 패킷을 교환하여야 함
  - 세션 유지를 위한 FrED는 시스템 허트비트로 사용되며 Data부는 '0'의 값을 가짐
- 최대 허트비트 지속시간 내에 패킷수신이 이루어지지 않으면 서버와 클라이언트에 의해 연결된 세션은 즉각 종료되어야 함

## 4) 세션종료

### (가) 세션종료 절차

- 세션의 종료는 서버의 종료요청 또는 클라이언트의 로그아웃 요청에 의해 발생되며 그 절차는 다음과 같음



<그림 6> 세션종료 절차

### (나) 종료요청

- 세션은 서버에 의해 능동적으로 종료될 수 있음
- 서버는 클라이언트와의 세션을 종료하기 위해서 종료사유를 포함한 종료요청 패킷을 전송할 수 있으며 사용되는 종료사유는 다음과 같음
  - 서버의 요청, 클라이언트의 요청, 서버종료(shutdown), 클라이언트종료(shutdown), 서버오류, 클라이언트 오류

#### (다) 로그아웃 요청

- 클라이언트는 세션을 종료하고자 하거나 서버로부터 세션종료를 요청받았을 경우 로그아웃 사유를 포함한 로그아웃 패킷을 전송하여 세션종료를 요청할 수 있음
- 사용되는 로그아웃 사유는 종료요청 사유와 동일함

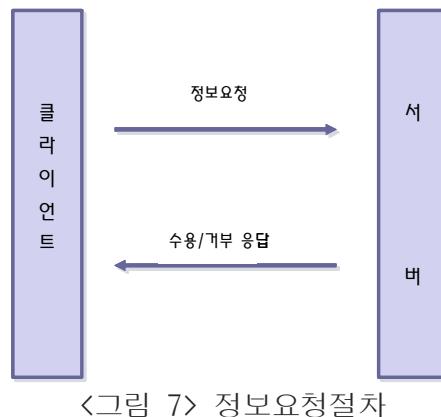
#### (라) FrED

- 로그아웃 요청을 전송받은 서버는 이에 대한 응답으로 FrED를 전송하고 세션을 종료함
  - 세션 종료단계의 FrED 패킷은 헤트비트 패킷이 아닌 로그아웃 요청에 대한 응답패킷으로 Data부의 값은 대응되는 패킷번호와 동일

### 5) 정보요청

#### (가) 정보요청 절차

- 세션이 설정된 후 클라이언트가 서버에 정보를 요청하는 과정은 다음과 같음



#### (나) 정보요청

- 클라이언트는 서버에 교통정보를 요청하기 위하여 서브스크립션(subscription: 교통정보교환 단계와 관련된 요청데이터패킷)을 전송함
  - 클라이언트는 서브스크립션에 요청정보를 명시하여 해당 기본교통정보 전송 가능
- 클라이언트는 서브스크립션에 정보요청방법과 정보제공방법을 명시하여 정보를 요청할 수 있음

- 정보요청에는 정보를 한번만 제공받기를 요청하는 “단일요청방법”과 일정기간 동안 지속적으로 정보를 제공받기 위한 “등록된 요청방법”이 있음
- 정보제공방법은 클라이언트가 결정할 수 있으며 데이터패킷에 의한 정보제공방법과 파일에 의한 정보제공방법 중 하나 선택 가능

#### (다) 수용/거부 응답

- 서버는 서브스크립션 정보의 유효 여부를 판단하여 “수용(accept)” 또는 “거부(reject)” 응답 패킷을 클라이언트로 전송함

### 6) 정보공개

#### (가) 정보제공 절차

- 정보가 요청되었을 경우 이에 대한 정보 공개 과정은 다음과 같음



<그림 8> 정보제공절차

#### (나) 정보제공

- 서버는 요청된 정보를 공개하기 위하여 요청된 정보가 담긴 퍼블리케이션 (publication) 패킷을 클라이언트로 전송함
  - 데이터 패킷으로 요청받은 경우, 퍼블리케이션 패킷에 요청된 정보 포함
  - 파일에 의해 정보제공을 요청받은 경우, 요청된 정보를 파일로 저장하고 파일의 이름과 파일 위치정보 제공
  - 데이터패킷으로 정보제공을 요청받은 경우, 생성된 데이터그램이 최대 데이터그램의 크기를 초과할 경우는 요청된 정보를 파일로 저장하고 파일의 이름과 파일 위치정보 제공

#### (다) 수용/거부응답

- 클라이언트는 수신된 퍼블리케이션의 유효여부를 판단하여 수용/거부 응답 패킷을 전송하여야 함
  - 클라이언트가 수신한 퍼블리케이션이 부적절하게 인코딩되었다고 판단된 경우, “거부” 데이터 패킷을 발생시켜야 하나 datexPublish-Guaranteed-bool 항목(퍼블리케이션 수신을 확인시키기 위한 응답데이터 전송여부)이 “true” 값을 가진 경우라면 “수용” 데이터 패킷 전송
  - 퍼블리케이션이 유효하고 datexPublish-Guaranteed-bool 항목이 “false” 값을 가질 경우, 서버로 응답패킷을 보내지 않음

#### (라) 파일전송

- 패킷이 아닌 파일로 정보를 제공할 경우 클라이언트는 제공받은 파일이름과 파일 위치정보를 이용하여 파일을 전송받음
  - 파일전송은 서브스크립션과 퍼블리케이션 데이터 패킷에서 결정된 파일전송 방법으로 수행(tftp, ftp 등)
  - 서버는 클라이언트로부터 전송완료 응답을 받을 때까지 파일이 다운로드 될 수 있도록 유지

#### (마) 전송완료

- 클라이언트는 파일전송이 완료되면 파일전송이 완료되었다는 것을 전송완료패킷으로 서버에 통보하여야 함

#### (바) FrED

- 서버는 파일전송완료패킷에 대한 응답으로 FrED 패킷을 전송하며 이때 FrED 패킷은 세션유지를 위한 허트비트패킷이 아닌 응답패킷으로 대응되는 패킷의 번호가 FrED의 값이 됨

## 2.6 정보센터 Adress(ID)부여

### 2.6.1 정보센터ID

- 센터 간 정보교환을 위한 센터 ID는 각 센터를 식별하기 위해 부여하는 고유인식부호로 총 8자리의 숫자로 표현하는 것을 기본으로 함
  - 로그인 정보 중 센터의 고유한 이름 사용 가능

### 2.6.2 정보센터ID 부여방법

- 센터 ID는 센터가 소재하는 행정구역의 우편번호(개정이전 6자리 우편번호)에 두 자리 확장자를 두는 것을 원칙으로 함
  - 이 원칙을 적용하지 아니할 경우에는 왼쪽 정렬을 기준으로 하며 오른쪽으로 8 자리까지 숫자 “0”을 채움



<그림 9> 정보센터 ID 부여 방법

- 두 자리 확장자는 행정구역내 2개 이상의 센터가 설치·운영되는 경우에 부여하는 식별자로서 원칙적으로 관할지역 경계의 남/서 지역부터, 북/동 지역으로, 도로의 상위등급으로부터 하위등급으로 순차적으로 부여함

<표 6> 정보센터 ID 부여 (예)

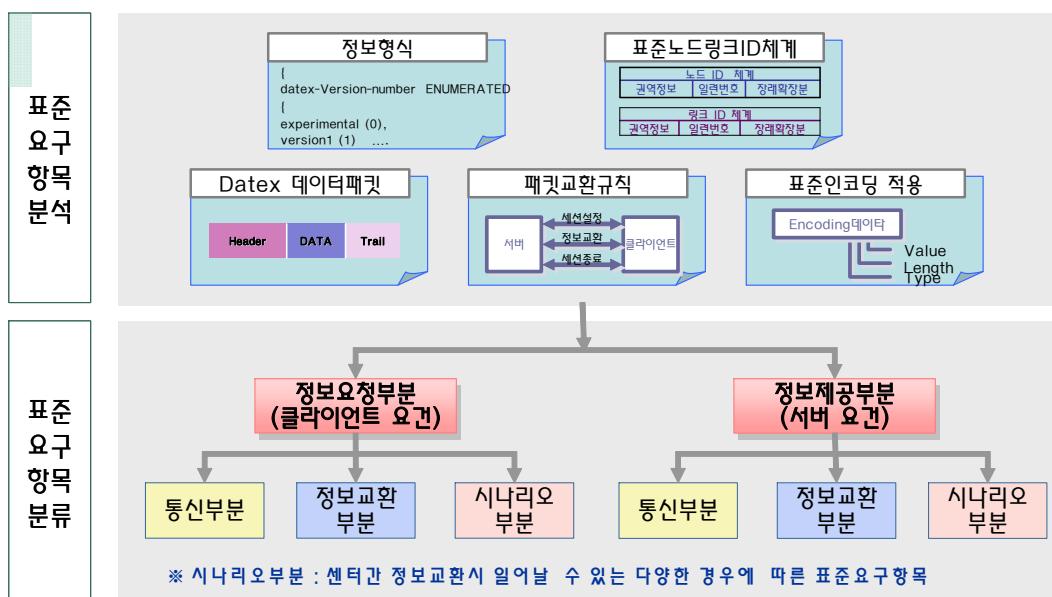
단체명	ID	단체명	ID
국토교통부 교통정보종합센터	42772100	한국도로공사 교통정보센터	46347000
서울특별시청 교통정보센터	10074400	성남시청 교통정보센터	46170000
강릉시청 교통정보센터	21070300	수원시청 교통정보센터	44270100
경기도청 교통정보센터	44170100	안산시청 교통정보센터	42570200
과천시청 교통정보센터	42771400	안양시청 교통정보센터	43172800
광주광역시청 교통정보센터	50270200	울산광역시청 교통정보센터	68070100
대구광역시청 교통정보센터	70071400	인천광역시청 교통정보센터	40575000
대전광역시청 교통정보센터	30278900	전주시청 교통정보센터	56170000
부산광역시청 교통정보센터	61173500	제주시청 교통정보센터	69070100
부천시청 교통정보센터	42073600	충주시청 교통정보센터	38070000

## 2.7 기술기준 준수여부 확인을 위한 검사방법

- ITS 시스템은 기술기준을 적용하여 구축하여야 하며, 기술기준을 올바르게 적용했는지 객관적으로 확인하는 검사방법도 마련되어 있음

### 2.7.1 검사 항목 도출

- 「기본교통정보교환 기술기준」 및 「대중교통(버스) 정보교환 기술기준」은 센터 간 정보교환을 위한 표준정보형식, 표준노드링크ID, 통신인터페이스 등을 규정하고 있으며 요청된 정보를 제공하기 위한 서버기능과 타 센터에 정보를 요청하여 정보를 수집하기 위한 클라이언트의 기능을 모두 만족하도록 요구하고 있음
- 따라서 기술기준 적용 적합성 시험 표준에서는 시험대상이 교통정보교환 기술기준 요구사항에 적합하게 정보를 교환하는지를 시험하기 위하여 규정된 방법에 따라 서버기능을 테스트하는 정보제공시험, 클라이언트기능을 테스트하는 정보요청시험, 그리고 시험대상의 기능을 종합적으로 시험하기 위한 통합시험으로 분류하여 시험항목을 구성함
- 정보제공시험과 정보요청시험은 각각 정보교환을 위한 기본 기능시험과 정보제공/교환 기능시험, 다양한 시나리오에 따른 시험으로 세분화됨



<그림 10> 기본교통정보교환 기술기준 표준준수검사 항목

## 2.7.2 검사 방법

- 「기본교통정보교환 기술기준 적용 적합성 시험 표준(ITSK-00031:2009)」, 「대중교통(버스)정보교환 기술기준 적용검증 시험 표준(ITSK-00040:2010)」에서는 각 시험항목에 따라 시험 방법 및 평가방법을 제시하고 있으며 시험은 ITS 표준화전담기관의 표준검증시스템에 해당 시스템을 연결하여 기본시험과 통합시험을 수행함

### 1) 기본검사(1차 검사)

- 표준검증시스템과 시험을 받고자 하는 대상 시스템을 연결하여 정보를 제공하는 서버기능과 정보를 요청하는 클라이언트기능으로 구분하여 기술기준에서 규정한 방법에 따라 정보를 교환하고 분석하는지를 검사함

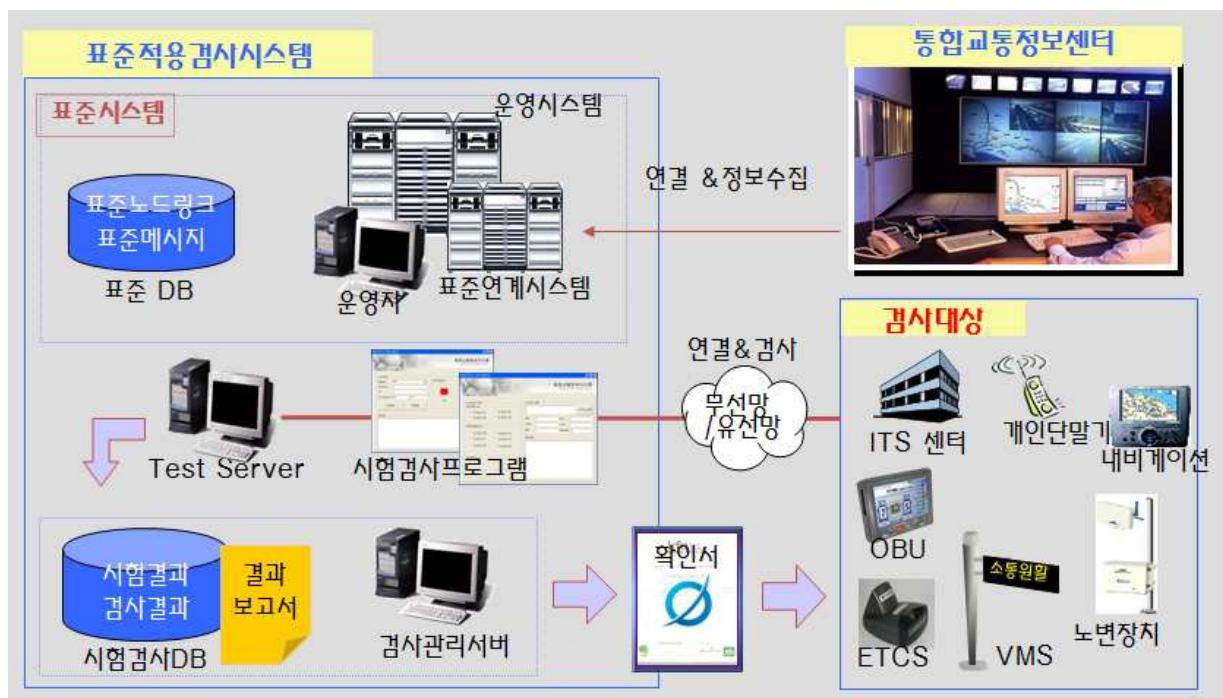
### 2) 통합검사(2차 검사)

- 시험을 받고자 하는 대상시스템의 특성을 고려하여 정해진 기간 동안 표준검증시스템을 연결하고 시험하여 실제 시스템 운영상에서 발생될 수 있는 다양한 환경 속에서의 정보교환의 성공률 및 정확성에 대해 시험함



<그림 11> ITS 기술기준 준수 여부 검사방법

- 정확한 표준준수여부 확인검사 결과를 얻기 위해서는 검증된 표준시스템이 검사 대상 시스템과 연결하여 시험이 이루어져야 함
- 검사용 데이터는 교통정보센터로부터 수집된 실제 정보를 사용하여 현실적인 검사환경에서 실시될 수 있도록 유지함
- ITS 표준화전담기관에서는 정확하고 객관적으로 표준준수여부 확인 검사를 위해 표준시스템을 구축하고, 시험표준에 따라 시험검사프로그램을 개발함



<그림 12> ITS 표준준수 여부 확인 검사방법

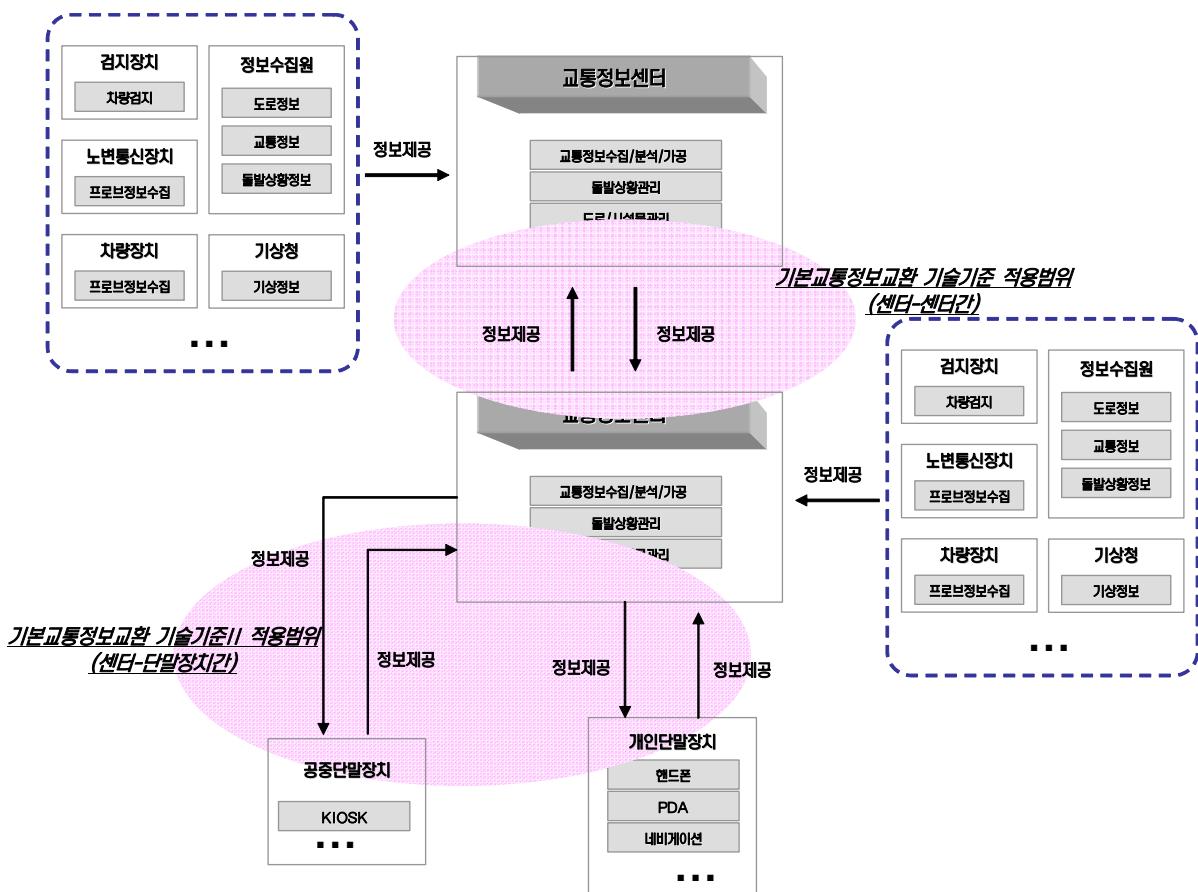
### **3. 기본교통정보교환 기술기준 II**

#### **3.1 개요**

- 센터-단말장치 간 기본교통정보교환 기술기준을 위한 표준은 의견수렴과 기술의 적합성 평가를 통해 「기본교통정보교환 기술기준Ⅱ」로 국토해양부(現 국토교통부)에서 제정·고시함
  - 제정 2006년 5월 30일, 국토해양부 고시 제2006-175호
  - 최종개정 2016년 4월 15일, 국토교통부 고시 제2016-207호 (일부개정)
- 본 장에서는 기술기준의 적용범위 및 연계체계를 설명하여 ITS 사업을 구상, 계획하는 단계에서 추진하는 정보연계사업이 기술기준을 준수해야하는지를 명확히 판단할 수 있도록 함
- 또한 정의된 정보의 활용방법과 세부항목의 구성 및 입력방법 등에 대해 구체적으로 제시하며, 완성된 정보를 센터와 단말장치 간에 교환할 수 있도록 정보교환을 위한 패킷구성, 인코딩 방법, 절차, 패킷 구성필드의 사용방법 등에 대해 함께 기술하도록 함

#### **3.2 기술기준 구성 및 적용범위**

- 본 기술기준은 검지장치, 정보수집원 등 또는 기본교통정보교환 기술기준에 의해 타 센터로부터 수집된 교통정보를 단말장치로 제공하거나 단말장치로부터 정보 수집 시 적용할 수 있음
  - 국가 ITS 아키텍처에서는 정보의 흐름을 센터에서 단말장치로 정의하고 있으며, 단말장치는 정보를 수신하여 이용자에게 제공
  - 고속도로순찰대원이 단말장치를 통해 센터로 정보를 제공하는 등 실제로는 단말장치에서 센터로 정보 제공
  - 기술기준에서는 국가 ITS 아키텍처와 실제 운영상황을 고려하여 센터-단말장치 간 정보교환을 정의



<그림 13> 기본교통정보교환 기술기준Ⅱ 적용범위

- 기술기준에서 정의하는 단말장치는 공중단말장치와 개인단말장치로 한정하며, 표출장치인 VMS는 제외함
  - 공중단말장치 : 터미널, 역사 등 공공장소에 설치되는 KIOSK
  - 개인단말장치 : 운영 및 관리목적으로 국가통합교통체계효율화법에서 정의한 사업시행자가 ITS사업의 일환으로 개발·구축하는 PDA 등
  - 기타 사업시행자가 대국민서비스를 위해 교통정보를 제공하는 장치
- 기술기준은 센터-단말장치 간 정보교환을 위해 정보형식과 이를 교환하기 위한 데이터 인터페이스로 구성됨

### 3.3 정보의 구성

- 센터-단말장치 간에 교환되는 정보는 다음의 4개 정보로 구성됨

<표 7> 기본교통정보교환 기술기준Ⅱ의 정보항목

ID	정보명	설명	정보주기	
			단방향제공	양방향제공
301	교통소통정보	도로의 소통상황정보를 제공하기 위한 정보항목으로 현재의 링크정보와 구간 예측소통정보를 제공할 수 있음	상시	요청시
302	교통통제정보	퍼레이드, 공사 등 계획된 이벤트에 의해 교통흐름에 영향을 주거나 도로통제가 발생될 경우 제공되는 정보항목	상황발생시	요청시 /상황발생시
303	돌발상황정보	사고, 화재 등 예측하지 못한 돌발상황 발생시 제공되는 정보항목	상황발생시	요청시 /상황발생시
304	도로상태정보	기후, 포장, 낙석 등에 의해 도로노면상태의 변화가 발생하였을 경우 제공되는 정보항목	상황발생시	요청시 /상황발생시

- 센터-단말장치 간 정보교환은 유선망, 라디오주파수, 무선망 등 다양한 통신망을 통해 이루어지고 있음
- 이에 유선망, 라디오주파수 등 통신비용에 영향을 받지 않는 경우에는 요청이 없어도 상시 제공할 수 있도록 교환 주기를 정의함
- 반면, CDMA망 등 통신비용에 영향을 미치는 경우는 센터 또는 단말기로부터 요청이 있을 경우에만 제공하며, 교통통제정보, 돌발상황정보, 도로상태정보의 경우는 운영방법에 따라 발생시에는 무조건적으로 제공할 수 있음

### 3.4 교통정보세항목 구현방법

#### 3.4.1 교통소통정보

- 교통소통정보는 정보생성시간, 링크식별번호, 링크속도를 제외한 기타 항목을 선택 항목으로 처리하여 정보제공전략, 목적 등에 적합하게 구성하여 적용할 수 있도록 정의됨

## 301 교통소통정보

본 정보항목은 도로의 교통상태 또는 예측 정보를 규약함

정보명	교통소통정보
아키텍처상 정보명	교통정보, 고속도로교통정보, 도시부 간선도로교통정보, 국도/지방도 교통정보
정보세부항목	속도, 교통량, 밀도, 통행시간, 대기길이, 점유율, 지체도, 구간예측평균속도, 구간예측통과시간, 추가정보
설명	도로의 교통소통상황 및 예측정도를 나타내는 정보

```

TRINFORLINKST DEFINITIONS AUTOMATIC TAGS::=BEGIN
TrafficCurrentLinkState ::= SEQUENCE
{
    trnt--MessageCreativeGeneralizedTime VisibleString,
    link--LinkIdNumber UTF8String (SIZE(1..40)),
    link--SpeedRate INTEGER (0..300),
    link--VolumeRate INTEGER (0..100000) OPTIONAL,
    link--DensityRate INTEGER (0..2000) OPTIONAL,
    link--TravelTimeQuantity INTEGER (0..10800) OPTIONAL,
    link--DelayQuantity INTEGER (0..12000) OPTIONAL,
    tfdt--VehicleQueueLengthQuantity INTEGER (0..500000) OPTIONAL,
    tfdt--OccupancyPercent INTEGER (0..100) OPTIONAL,
    pctt--PredictionAverageLinkSpeedRate INTEGER (0..300) OPTIONAL,
    pctt--PredictionTravelTimeQuantity INTEGER (0..10800) OPTIONAL,
    pctt--PredictionTimeGeneralizedTime VisibleString OPTIONAL,
    pctt--AdditionalInformationText UTF8String (SIZE(1..256)) OPTIONAL
}
END

```

정보의 이름/시작

- ① 정보생성시간
- ② 링크식별번호
- ③ 링크속도
- ④ 링크교통량
- ⑤ 링크밀도
- ⑥ 링크통행시간
- ⑦ 링크지체
- ⑧ 차량대기행렬길이
- ⑨ 검지기점유율
- ⑩ 구간예측평균속도
- ⑪ 구간예측통과시간
- ⑫ 예측시간
- ⑬ 소통정보의 추가입력

정보의 끝

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTCTime)	날짜, 시각
링크식별번호	네트워크내 링크에 대한 고유식별번호	번호
링크속도	주어진 시간동안에 링크를 주행한 차량의 평균 운행속도	km/시
링크교통량	주어진 시간동안에 링크를 통과한 차량	대/시
링크밀도	링크의 단위 킬로미터당 차량수	대/시
링크통행시간	주어진 시간동안에 링크를 주행한 차량의 평균통행시간	초
링크정체	특정 링크를 통행하는 차량의 정체시간	초
차량대기행렬길이	검지기에 의해 측정된 차량들의 평균대기행렬 길이	m
검지기점유율	검지기 점유 백분율 (검지된 시간/총 단위시간×100)	%
구간예측평균속도	해당구간 예측되는 평균속도	km/시
구간예측통과시간	해당구간의 예측되는 차량의 평균통행시간	초
예측시각	예측정보가 적용되는 해당시각	날짜, 시각
소통정보의 추가입력	부가정보의 입력	문자

- ①정보생성시간은 소통정보가 발생된 시간을 의미하며 날짜 및 시간 조합의 "YYYYMMDDhhmmss" 형태로 입력할 수 있음
  - Y(년도), M(월), D(일자), h(시간), m(분), s(초)를 의미하며 VisibleString형태로 입력
  - 모든 시각항목은 UTC(Universal Time Coordinated, 협정 세계시) 사용
- ②링크식별번호는 위치를 나타내기 위한 항목으로 「지능형교통체계 표준 노드 링크 구축·관리지침」(국토교통부 고시 제2015-755호(일부개정))의 노드/링크 ID 부여방법에 따라 정해진 링크 ID를 사용함
- ③링크속도는 ②링크식별번호, 즉 해당 링크를 평균 몇 km의 속도로 통과하는지를 나타내는 교통소통의 지표값으로 해당 링크를 일정시간동안 주행한 모든 차량의 평균속도인 시간평균속도를 사용하며 입력값의 단위는 km/h로 함
- ④링크교통량, ⑤링크밀도, ⑥링크통행시간, ⑦링크지체는 각각 주어진 시간동안에 링크를 통과한 차량대수, 단위 킬로미터당 차량수, 일정시간동안 링크를 주행한 차량의 평균통행시간, 링크통행차량의 지체시간으로 입력단위는 링크교통량과 링크밀도는 대/시, 링크통행시간과 링크지체시간은 초를 사용함
- ⑧차량대기행렬길이와 ⑨검지기점유율은 검지기에 의해 측정된 값으로 차량대기행렬길이는 미터, 점유율은 백분율을 사용함
- ⑩구간예측평균속도, ⑪구간예측통과시간, ⑫예측시간은 해당 링크에 대한 예측정보로써 선택사항으로 정의되어 있으나 ⑩ 또는 ⑪ 항목을 사용할 경우에는 예측정보에 대한 시간을 나타내는 ⑫예측시간을 필수항목으로 처리하여야 함
- ⑬소통정보의 추가입력은 교통소통정보에 대한 추가적 정보를 제공하기 위한 항목으로 자체 시 자체의 원인정보제공, 위험요소발생 시 위험요소에 대한 정보제공 등으로 사용할 수 있으며 추가정보는 최대 256 글자를 넘지 않음

### 3.4.2 교통통제정보

#### 302 교통통제정보

본 정보항목은 도로의 이벤트에 따른 교통통제정보의 정보세항목을 규약함

정보명	교통통제정보
아키텍처상 정보명	교통통제정보, 도로정보
정보세부항목	위치, 통제유형, 대상, 시간
설명	미리계획된 도로의 공사, 행사 등 교통의 통제상황을 나타내는 이벤트 정보

```

TRINFOEVENTIDENTITY DEFINITIONS AUTOMATIC TAGS ::= BEGIN
TrafficEventInfo ::= SEQUENCE
{
    trnt-MessageCreativeGeneralizedTime VisibleString,
    evnt-LocationNodeIdNumber SEQUENCE OF UTF8String,
    evnt-LocationLinkIdNumber SEQUENCE OF UTF8String,
    evnt-LocationTypeCode ENUMERATED OPTIONAL,
    evnt-LaneBlockedOrClosedTypeCode ENUMERATED OPTIONAL,
    evnt-LanesBlockedOrClosedNumber INTEGER (0..255) OPTIONAL,
    evnt-LocationBeginNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    evnt-LocationLinearDistanceOffsetBeginNumber INTEGER (0..99999) OPTIONAL,
    evnt-LocationEndNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    evnt-LocationLinearDistanceOffsetEndNumber INTEGER (0..99999) OPTIONAL,
    evnt-LocationAdditionalInfoText UTF8String (SIZE(1..256)) OPTIONAL,
    evnt-DescriptionTypeEventCode ENUMERATED OPTIONAL,
    evnt-DescriptionTypeSpecialEventCode ENUMERATED OPTIONAL,
    evnt-DescriptionTypeEventOtherText UTF8String (SIZE(1..256)) OPTIONAL,
    evnt-ResponseAlternateRouteText UTF8String (SIZE(1..1024)),
    evnt-LanesShouldersBlockedCode ENUMERATED OPTIONAL,
    evnt-TimelinestarGeneralizedTime VisibleString OPTIONAL,
    evnt-TimelineEndGeneralizedTime VisibleString OPTIONAL,
    evnt-TimelineDurationTime INTEGER (0..4294967295) OPTIONAL
}
END

```

정보의 이름/시작

- ① 정보생성시각
- ② 이벤트위치노드
- ③ 이벤트위치링크
- ④ 이벤트위치유형
- ⑤ 차단차로유형
- ⑥ 차단차로수
- ⑦ 이벤트시작노드
- ⑧ 위치선형거리옵션시점
- ⑨ 이벤트종점노드
- ⑩ 위치선형거리옵션종점
- ⑪ 이벤트위치부연설명
- ⑫ 이벤트유형구분
- ⑬ 특별이벤트유형설명
- ⑭ 이벤트부연설명
- ⑮ 대안경로안내
- ⑯ 차선길어깨차단코드
- ⑰ 이벤트개시시각
- ⑱ 이벤트종료시각
- ⑲ 이벤트지속표준시

정보의 끝

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTCTime)	날짜, 시각
이벤트 위치노드	도로 이벤트가 해당되는 모든 노드번호	-
이벤트 위치링크	도로 이벤트가 해당되는 모든 링크 번호	-
이벤트 위치유형	이벤트 통제위치의 유형	-
차단차로유형	이벤트로 인한 차로 차단의 유형	-
차단차로수	도로 이벤트로 인해 차단/통제된 차로수	수
이벤트 시작노드	도로 혹은 링크 중간에서 이벤트의 시작특정지점을 표시할 경우 진행방향의 바로 전 노드	번호
위치선형거리옵션 시점	이벤트 시작노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	1m
이벤트 종점노드	도로 혹은 링크 중간에서 이벤트의 종료특정지점을 표시할 경우	번호

명칭	설명	단위
	진행방향의 바로 다음 노드	
위치선형거리옵셋 종점	이벤트 종점노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	1m
이벤트 위치부연설명	도로 이벤트의 위치에 대한 부연설명(문자)	문자
이벤트유형구분	각종 도로 이벤트의 유형을 구분할 수 있는 코드	-
특별이벤트유형설명	도로 이벤트 중 특별한 도로 이벤트의 유형을 표현하기 위한 코드	-
이벤트 부연설명	이벤트에 대한 기타사항, 부연설명	문자
대안경로안내	통제된 도로에 대해 효율적으로 운영할 수 있는 대안경로의 안내	문자
차선길어깨차단코드	도로 이벤트로 인해 차단된 길어깨를 나타내는 코드	-
이벤트개시시각	특별한 도로 이벤트의 실제적인 개시 시각	날짜, 시각
이벤트종료시각	도로 이벤트의 실제적인 종료 시간	날짜, 시각
이벤트지속표준시	도로 이벤트의 실제적인 진행 시간	분

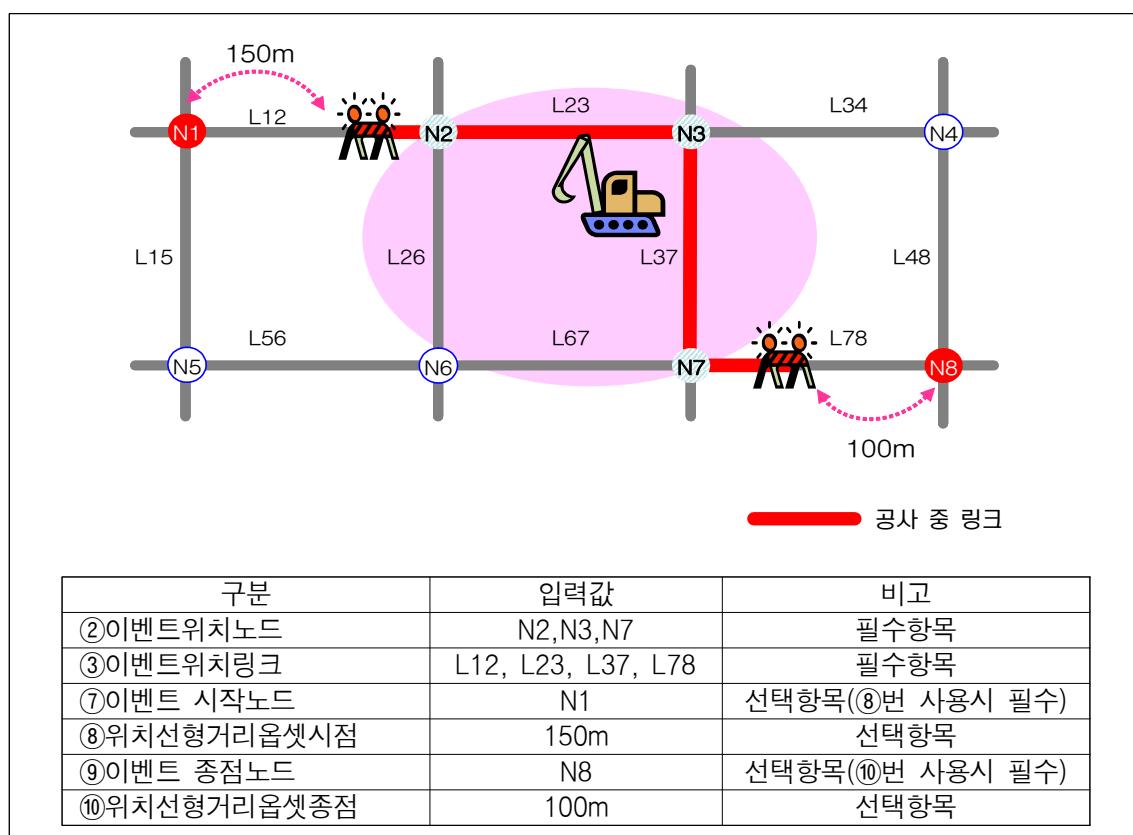
- 교통통제정보는 크게 통제방법 또는 유형, 이벤트발생위치, 시간정보, 부연설명으로 구성되어 있음
- 먼저 통제유형을 위한 항목 및 코드값은 다음의 표와 같으며, 모두 선택사항으로 정의되어 운영, 또는 정보제공 상 필요한 항목을 선택하여 사용할 수 있음

<표 8> 기본교통정보교환 기술기준Ⅱ의 통제유형의 주요 코드값

통제유형구분	통제코드값
④이벤트위치유형	해당사항없음(0), 지점통제(1), 구간(링크)통제(2), 지구(구역, 행정구획)통제(3), 교차로통제(4)
⑤차단차로유형	전면통제(0), 부분통제(1)
⑥차단차로수	도로이벤트로 인해 차단/통제된 차로수
⑫이벤트유형구분	없음(0), 기타(1), 공사(2), 도로폐쇄(3)
⑯특별이벤트유형설명	없음(0), 기타(1), 퍼레이드(2), 스포츠이벤트(3), 시위 및 집회(4), 축제(5), 예술공연(6), 수행행렬(7)
⑯차선길어깨차단코드	없음(0), 기타(1), 우측길어깨(2), 좌측길어깨(3), 양쪽길어깨(4)

- 본 기술기준에서 정보의 위치는 표준노드링크 ID로 표현되며, 교통통제 위치의 정확한 표현을 위해서 교통통제정보에서는 표준노드링크ID를 이용한 다양한 이벤트 발생위치표현항목이 포함되어 있음

- ②이벤트위치노드 : 도로 이벤트에 해당되는 모든 노드번호로 표준노드ID 적용
  - ③이벤트위치링크 : 도로이벤트가 해당되는 모든 링크번호로 표준링크ID 적용
  - ⑦이벤트 시작노드 : 도로 혹은 링크 중간에서 이벤트의 시작 특정지점을 표시할 경우 진행방향의 바로 전 노드로 표준노드ID 적용
  - ⑧위치선행거리옵셋시점 : 이벤트 시작노드에서 거리로써 측정된 도로 이벤트 발생 구간 시점의 정확한 위치로 단위는 미터(m)를 사용함
  - ⑨이벤트 종점노드 : 도로 혹은 링크 중간에서 이벤트의 종료특성지점을 표시할 경우 진행방향의 바로 다음 노드로 표준노드ID 적용
  - ⑩위치선행거리옵셋종점 : 이벤트 종점노드에서 거리로써 측정된 도로 이벤트 발생 구간 종점의 정확한 위치로 단위는 미터(m)를 사용함
- 위치표현 시 ⑦이벤트 시작노드, ⑨이벤트 종점노드는 선택항목이나 ⑧위치선행거리옵셋시점, ⑩위치선행거리옵셋종점시 사용 시 반드시 사용하여야 함
- 다음 그림은 위치항목 값 입력방법의 예임



<그림 14> 기본교통정보교환 기술기준Ⅱ의 교통통제발생 예시

- 교통통제의 시간정보 제공을 위한 항목은 다음 표와 같이 활용할 수 있음

<표 9> 기본교통정보교환 기술기준Ⅱ의 교통통제 시간정보항목

구분	단위	설명	비고
①정보생성시간	YYYYMMDDhhmmss	교통통제정보를 제공하는 시간	필수항목
⑦이벤트개시시각	YYYYMMDDhhmmss	이벤트의 실제적인 개시시각	선택항목
⑧이벤트종료시각	YYYYMMDDhhmmss	이벤트의 실제적인 종료시각	선택항목
⑨이벤트지속표준시간	분	이벤트 지속시간으로 ⑧에서 ⑦를 뺀 값	선택항목

### 3.4.3 돌발상황정보

#### 303 돌발상황정보

본 정보항목은 도로의 돌발상황발생정보를 규약함

정보명	돌발상황처리요청정보
아키텍처상 정보명	돌발상황처리요청, 돌발상황정보, 돌발상황발생정보, 기본정보
정보세부항목	위치, 시각, 돌발상황 유형, 돌발상황상태, 정보갱신상태
설명	예상하지 못한 도로상의 유고상황을 알려주는 정보로서 상시 제공받는 정보가 아니라 상황발생시 제공되는 정보

```

TRINFOINCIDENTCONDITIONS DEFINITIONS AUTOMATIC TAGS ::= BEGIN
TrafficIncidentConditions ::= SEQUENCE
{
  trnt--MessageCreativeGeneralizedTime VisibleString,
  node--NodeldNumber UTF8String (SIZE(1..40)),
  link--LinkIdNumber UTF8String (SIZE(1..40)),
  orgn--ContactOrganizationNameText UTF8String (SIZE(1..128)) OPTIONAL,
  evnt--DescriptionTypeIncidentCode BIT STRING OPTIONAL,
  evnt--DescriptionTypeIncidentOther UTF8String (SIZE(1..256)) OPTIONAL,
  evnt--LocationBeginNodeldNumber UTF8String (SIZE(1..40)) OPTIONAL,
  evnt--LocationLinearDistanceOffsetBeginNumber INTEGER (0..99999) OPTIONAL,
  evnt--LocationEndNodeldNumber UTF8String (SIZE(1..40)) OPTIONAL,
  evnt--LocationLinearDistanceOffsetEndNumber INTEGER (0..99999) OPTIONAL,
  evnt--UpdateGeneralizedTime VisibleString OPTIONAL,
  evnt--IncidentStatusCode BIT STRING OPTIONAL,
  evnt--IncidentStatusOther UTF8String (SIZE(1..256)) OPTIONAL,
  evnt--LanesBlockedOrClosedNumber INTEGER (0..255) OPTIONAL,
  evnt--ResponseAlternateRouteText UTF8String (SIZE(1..1024)),
  evnt--UpdateTypeCode BIT STRING OPTIONAL,
  evnt--UpdateTypeOtherText UTF8String (SIZE(1..256)) OPTIONAL
}
END

```

정보의 이름/시작

- ① 정보생성시간
- ② 노드식별번호
- ③ 링크식별번호
- ④ 연락기관명칭
- ⑤ 돌발상황의유형설명
- ⑥ 돌발상황유형부연설명
- ⑦ 이벤트시작노드
- ⑧ 위치선형거리옵션시점
- ⑨ 이벤트종점노드
- ⑩ 위치선형거리옵션종점
- ⑪ 돌발상황갱신시각
- ⑫ 돌발상황상태
- ⑬ 돌발상황상태부연설명
- ⑭ 차단차로수
- ⑮ 대안경로안내
- ⑯ 갱신유형코드
- ⑰ 갱신유형부연설명

정보의 끝

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTCTime)	날짜, 시각
노드식별번호	네트워크 내 노드에 대한 고유식별번호	번호
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
연락기관명칭	지역 내의 특정 기관의 명칭. 상호관련된 기관이 있을 경우 최상위 기관을 명시	문자
돌발상황의 유형설명	돌발상황에 대한 유형을 표현하기 위한 코드	-
돌발상황유형의 부연설명	돌발상황의 유형에 대한 부연설명	문자
이벤트시작노드	도로 혹은 링크 중간에서 이벤트의 시작특정지점을 표시할 경우 진행방향의 바로 전 노드	번호
위치선형거리옵셋 시점	이벤트 시작노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	1m
이벤트종점노드	도로 혹은 링크 중간에서 이벤트의 종료특정지점을 표시할 경우 진행방향의 바로 다음 노드	번호
위치선형거리옵셋 종점	이벤트 종점노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	1m
돌발상황갱신시각	특별한 도로 이벤트에 대하여 최근 변경 시 갱신되어진 시각	날짜, 시각
돌발상황상태	돌발상황 상태를 나타내는 코드	-
돌발상황상태의 부연설명	돌발상황 상태의 부연설명	문자
차단차로수	도로 이벤트로 인해 차단/통제된 차로수	수
대안경로안내	통제된 도로에 대해 효율적으로 운영할 수 있는 대안경로의 안내	문자
갱신유형코드	특별한 도로 이벤트에 대해 정보의 갱신 유형을 나타내는 코드	-
갱신유형의 부연설명	정보의 갱신유형의 부연설명	문자

- 돌발상황정보에는 ①정보생성시간과 ⑪돌발상황갱신시간으로 돌발상황발생에 대한 시간표현이 가능함
- ①정보생성시간은 돌발상황이 발생된 최초의 시각으로 이를 기준으로 향후 갱신되는 돌발상황정보가 어떤 돌발상황에 대한 연장정보인지를 확인하는데 보조적 정보로 사용할 수 있어, 동일 돌발상황에 대한 정보생성시간은 변하지 않음
- 돌발상황의 진행상태 정보제공을 위해서 갱신정보를 전송할 때에는 ⑪돌발상황갱신시간으로 해당 정보의 시각을 표현할 수 있음
- 다음 표는 ⑫돌발상황상태에 따른 갱신유형코드값과 정보생성시간 및 갱신시간의 입력예를 나타냄

<표 10> 기본교통정보교환 기술기준Ⅱ의 돌발상황발생정보 시간항목 값 입력(예)

⑫돌발상황상태	⑯갱신유형코드 값	①정보생성시간 값	⑪돌발상황갱신시간 값
사고감지	신규	20060605100000	20060605100000
처리 및 복구중	업데이트	20060605100000	20060605101500
완료	삭제	20060605100000	20060605103000

- ②노드식별번호와 ③링크식별번호는 돌발상황이 발생된 구간을 표현하기 위한 항목으로 표준노드링크ID를 사용함
  - 데이터유형은 UTF8String으로 2개 이상의 노드 또는 링크에 걸쳐 발생되었을 경우는 구분자를 사용하여 데이터를 입력할 수 있음
  - 돌발정보발생위치를 정확하게 표시하기 위해서는 ⑦, ⑧, ⑨, ⑩ 항목의 시작노드, 종점노드, 옵셋거리를 활용하여 나타낼 수 있음
- ④연락기관 명칭은 상호 관련된 기관을 명시하기 위한 항목으로 돌발상황의 접수 또는 관할기관 중 최상위기관을 입력함

### 3.4.4 도로상태정보

#### 304 도로상태정보

본 정보항목은 도로의 노면상태에 대하여 규약함

정보명	도로상태정보
아키텍처상 정보명	도로정보
정보세부항목	노면상태, 강우/강설수위
설명	기상조건 및 돌발상황에 따른 도로의 상태정보

```

TRINFORDWCON DEFINITIONS AUTOMATIC TAGS ::= BEGIN
TrafficRoadwayConditions ::= SEQUENCE
{
  trnt--MessageCreativeGeneralizedTime VisibleString,
  node--NodeidNumber UTF8String (SIZE(1..40)),
  link--LinkIdNumber UTF8String (SIZE(1..40)),
  link--BeginNodeidNumber UTF8String (SIZE(1..40)) OPTIONAL,
  link--EndNodeidNumber UTF8String (SIZE(1..40)) OPTIONAL,
  link--LengthQuantity INTEGER (0..160000) OPTIONAL,
  link--LocationAdditionalInfoText UTF8String (SIZE(1..256)) OPTIONAL,
  link--SurfaceConditionCode BIT STRING OPTIONAL,
  link--LaneBlockedOrClosedCode BIT STRING OPTIONAL,
  evnt--LanesBlockedOrClosedNumber INTEGER (0..255) OPTIONAL,
  evnt--ResponseAlternateRouteText UTF8String (SIZE(1..1024)),
  evsn--DataSurfacePrecipitationDepth INTEGER (0..3000) OPTIONAL,
  evsn--DataSurfaceTemperatureQuantity INTEGER (0..300) OPTIONAL
}
END
  
```

정보의 이름/시작

- ① 정보생성시간
- ② 노드식별번호
- ③ 링크식별번호
- ④ 시점노드식별번호
- ⑤ 종점노드식별번호
- ⑥ 링크길이
- ⑦ 지역추가정보
- ⑧ 도로노면상태
- ⑨ 도로차단여부
- ⑩ 차단차로수
- ⑪ 대안경로안내
- ⑫ 강우/설량수위
- ⑬ 도로표면온도

정보의 끝

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTCTime)	날짜, 시각
노드식별번호	네트워크 내 노드에 대한 고유식별번호	번호

명칭	설명	단위
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
시점노드식별번호	링크 시점에 해당되는 노드의 고유식별번호	번호
종점노드식별번호	링크 종점에 해당되는 노드의 고유식별번호	번호
링크길이	시점노드에서 종점노드까지의 링크 길이	m
지역추가정보	이용자가 명확히 파악할 수 있는 ASCII코드로 된 지역정보	문자
도로노면상태	링크의 현재 도로노면 상태	-
도로차단여부	도로차단여부	-
차단차로수	도로 이벤트로 인해 차단/통제된 차로수	수
대안경로안내	통제된 도로에 대해 효율적으로 운영할 수 있는 대안경로의 안내	문자
강우/설량수위	환경감지기에서 노면의 강우량 수위를 나타내는 숫자적인 색인	mm
도로표면온도	환경감지기에서 섭씨로 측정된 도로의 표면 온도	°C

- 도로상태정보의 위치를 나타내기 위한 정보세항목으로는 ②노드식별번호와 ③링크식별번호가 있으며, 돌발상황정보와 같이 2개 이상의 노드 또는 링크에 대한 도로상태정보를 전송하기 위해서는 구분자를 사용하여 노드와 링크값을 입력할 수 있음
- ④시점노드식별번호와 ⑤종점노드식별번호는 시작위치와 종료위치를 나타낼 필요가 있을 경우 위치정보에 대한 보조적 정보로 사용할 수 있음
- ⑦지역추가정보는 도로상태정보가 발생된 위치에 대한 부가설명으로 지역명, 위치명, 교차로명 등을 사용할 수 있음
- ⑧도로노면상태는 기후 또는 돌발상황, 외부환경변화 등에 의해 변화된 노면상태정보를 제공하기 위한 항목으로, 중복선택이 가능하며 "노면습윤상태", "적설"의 경우는 ⑫강우/설량수위로 정보를 추가적으로 제공 가능
- ⑨도로차단여부는 노면상태에 의해 도로의 차단여부를 나타내기 위한 것으로 "차단안됨" 선택의 경우더라도 차로 일부분이 통제될 경우는 ⑩차단차로수를 활용하여 차로차단에 대한 추가정보를 제공할 수 있음
  - 만일 일부차로의 운행이 가능하더라도 차량의 증가, 위험요소 존재 등의 이유로 ⑪대안경로안내를 활용하여 차량우회 유도가 가능함

- 또한 차단으로 인해 도로의 이용이 불가능할 경우는 ⑪대안경로안내를 활용하여 차량을 우회시킬 수 있음

<표 11> 도로 및 차로차단 경우의 수에 따른 선택항목 입력(예)

구분	⑨도로 차단여부	⑩차단차로수	⑪대안경로안내
도로차단안함/차로차단안함	차단안됨(3)	“0” 또는 사용안함	필요시 입력
도로차단안함/차로일부차단	차단안됨(3)	차단차로수 입력	필요시 입력
도로차단/전차로 차단	차단됨(4)	사용안함	입력 권장

## 3.5 패킷구성 및 교환방법

### 3.5.1 패킷구조 및 입력방법

- 패킷은 Header, Data부, Tail부로 구성되어 있음

Header부				Data부	Tail부
Version No.	Authentication Info	Packet No	Header Option	PDU	Crc

<그림 15> 기본교통정보교환 기술기준Ⅱ의 기본패킷구조

- 본 패킷구조는 무선통신망, TCP/IP통신 방식 등을 위한 구조이며, 시리얼통신방식에서는 인코딩된 패킷에 시작제어코드인 STX(start of text) 1byte와 2byte 크기의 값이 없, 그리고 종료제어코드인 ETX(end of text) 1byte를 붙여 전송 할 수 있음

#### (가) datex-version-nbr

- 데이터패킷버전번호로 제정된 기술기준이므로 “1”의 값을 입력함

#### (나) datex-Authenticationinfo-txt

- 데이터부의 속성을 나타내기 위한 인증코드값으로 PDU에 따라 다음의 값을 입력함
  - Subscription : 0x01
  - Accept : 0x02
  - Reject : 0x03

- Publication : 0x40
  - TrafficInformationLinkState : 0x41
  - TrafficEventIdentity : 0x42
  - TrafficIncidentCondition : 0x43
  - TrafficRoadwayConditions : 0x44
- 인증코드값은 핵사값으로 1OCTET 크기의 OCTET STRING형태로 인코딩함

#### **(다) datex-Packet-nbr**

- 데이터패킷의 일련번호로 기존에 전송한 패킷과는 다른 새로 생성된 패킷임을 구분하기 위해 활용할 수 있음
  - 예를 들어 정보요청 시 동일한 정보요청패킷이 중복 전송된 것인지 별개의 정보 요청이 2회 전송된 것인지 등 파악 가능
- 또한 정보제공 후 응답패킷이 전송되었을 때 어떤 정보제공패킷에 대한 응답패킷인지를 확인하는데 사용됨

#### **(라) datex-SenderID/Dates-ReceiverID**

- 패킷을 송신 또는 수신하는 시스템의 고유번호로 송신 또는 수신시스템을 명확히 할 경우 선택적으로 사용할 수 있음

#### **(마) datex-DataPacket-time**

- 데이터패킷생성시간을 의미함
- 본 기술기준에서는 정보요청 시 한 개의 패킷으로 응답하는 것을 기본으로 하나 정보의 제공량이 많이 패킷을 분할하여 보낼 경우에는 datex-Packet-nbr값을 하나의 값으로 고정하고 데이터패킷생성시간을 달리함으로써 하나의 정보요청에 대한 일련의 정보제공패킷으로 처리할 수 있음

#### **(바) datex-Crc-nbr**

- 데이터의 오류를 검사하기 위한 check digit로 PDU부분의 인코딩 된 byte stream을 ISO 3309에서 정의된 CRC16 알고리즘을 사용하여 계산함
  - CRC16 계산식 = X<sub>16</sub> + X<sub>15</sub> + X<sub>2</sub> + 1

### 3.5.2 데이터부 구성

#### 1) 서브스크립션(Subscription)

##### (가) datexSubscription-Area-scope

- 교통정보의 지역적 제공범위로써 정보를 요청하는 시스템에서 정보제공범위를 지정할 수 있으며, 다음 항목 중 한 가지를 선택할 수 있음
  - local-Area : 일부구간에 대한 정보를 요청하기 위한 코드값으로 제공구간의 범위는 제공시스템에서 교통정보제공정책 또는 시스템 간 결정된 범위에 따라 제공
  - national-Area : 전국을 대상으로 정보를 요청하기 위한 코드값

##### (나) datexSubscription-Loacation-reference

- 위치참조를 나타내기 위한 정보항목으로 datexSubscription-Area-scope 코드값이 local-Area일 경우 반드시 사용해야 함
- 즉, 위치참조정보를 바탕으로 해당 위치정보로부터 일정구간의 정보를 제공할 수가 있으며, 위치참조는 표준노드·링크ID값을 사용함
  - 현재 구축·운영되고 있는 BIS시스템의 개인단말에서는 일반적으로 특정 정류장 ID를 입력값으로 대중교통(버스)정보를 요청 및 제공
  - 데이터인터페이스를 이용하여 개인단말로부터 대중교통(버스)정보를 요청할 경우에는 위치참조값으로 표준노드·링크ID가 아닌 정류장 ID를 사용 가능

##### (다) datexSubscription-Road-class

- 제공받고자 하는 도로의 종류를 선택하기 위한 정보항목으로 고속도로, 국도, 시군도, 지방도, 기타도로 중 증복선택이 가능함

##### (라) datexSubscription-Userinfo

- 제공정보가 유료이거나 정보요청 시스템의 인증이 필요할 경우 선택적으로 사용할 수 있는 항목으로, UserID와 비밀번호, 그리고 인증을 위한 추가적 ID로 구성됨

#### (마) datexSubscription-Pdu

- 요청하고자 하는 정보의 ID를 입력하기 위한 정보항목으로 한 개의 정보요청뿐 아니라 두 개 이상의 정보를 한 번에 요청할 수 있도록 SEQUENCE OF 형태로 정의

### 2) 퍼블리케이션(Publication)

#### (가) datexPublication-Response-existence

- 제공한 정보에 대한 응답여부를 결정하기 위한 정보항목으로 입력값에 따른 응답여부는 다음과 같음
  - TRUE : 정보를 제공받은 시스템은 이에 대한 응답패킷 전송
  - FALSE : 정보를 제공받은 시스템은 제공받은 정보에 대한 응답 생략

#### (나) datexPublication-Publication-Pdu

- 제공하고자 하는 교통정보가 포함된 정보항목으로 교통정보항목에 따라 해더부의 datex-Authentication-txt의 값이 결정됨
- 만일, 2가지 이상의 정보요청으로 하나의 퍼블리케이션에 서로 다른 2가지 이상의 제공정보가 포함될 경우 datex-Authentication-txt값은 퍼블리케이션을 의미하는 “0x40”을 입력함

### 3) 수용(Accept)

- 수용패킷의 데이터부는 수용패킷번호(datexAccept-Packet-nbr)와 수용형태(datexAccept-Type)으로 구성되어 있음
- 수용패킷번호는 응답대상인 수신된 패킷의 번호(datex-Packet-nbr)를 입력함
  - 만일 수신된 퍼블리케이션의 해더부에 있는 패킷의 번호가 “1”의 값을 갖는 경우 이에 대한 수용패킷 데이터부의 수용패킷번호는 “1”이 됨
  - 이는 수용패킷의 해더부에 있는 패킷번호와 무관
- 수용형태는 응답하고자 하는 대상 패킷의 종류를 선택하기 위한 항목으로 서브스크립션 또는 퍼블리케이션 중 어떤 패킷에 대한 응답인가를 나타냄

## 4) 거부(Reject)

- 거부패킷은 거부패킷번호(datexReject-Packet-nbr), 거부형태(datexReject-Type), 거부코드(datexReject-Code), 거부사유(datexReject-Reason)로 구성됨
- 거부패킷번호와 거부형태의 값은 수용패킷의 경우와 동일하게 입력함
- 거부코드의 경우는 거부의 원인이 되는 코드를 선택할 수 있으며, 거부사유 정보항목을 이용하여 세부원인정보를 추가로 제공할 수 있음
  - 거부사유 정보항목은 선택항목이나 거부 코드값으로 기타사항(others)이 선택되었을 경우는 거부의 원인 정보를 제공하기 위해 사용

### 3.5.3 데이터교환절차

- 정보요청, 정보제공, 응답패킷(승락, 거부)의 교환순서로 데이터교환절차가 정의되며, 통신비용 또는 정보제공방식에 따라 교환절차가 구성됨

<표 12> 통신망의 특성에 따른 데이터 교환절차

구분	단방향통신	양방향통신	
		통신비 제약 有	통신비 제약 無
통신망 종류	FM DARC, Passive DSRC 등	CDMA, 종량제무선망 등	TCP/IP, 정액제무선망 등
교환 절차	<p>서버 ← 정보요청 서버 → 요청에 대한 응답 서버 ← 정보제공 서버 → 제공에 대한 응답</p> <p>○ 필수사항 △ 선택사항 ✗ 사용안함</p>	<p>서버 ← 정보요청 서버 → 요청에 대한 응답 서버 ← 정보제공 서버 → 제공에 대한 응답</p> <p>○ 필수사항 △ 선택사항 ✗ 사용안함</p>	<p>서버 ← 정보요청 서버 → 요청에 대한 응답 서버 ← 정보제공 서버 → 제공에 대한 응답</p> <p>○ 필수사항 △ 선택사항 ✗ 사용안함</p>
	<ul style="list-style-type: none"> <li>-정보요청 없이 일방적 정보 제공</li> <li>-단방향이므로 요청 또는 제공에 대한 응답없음</li> </ul>	<ul style="list-style-type: none"> <li>-정보요청/필요시 정보제공</li> <li>-통제정보, 도로정보, 돌발상황정보의 경우</li> <li>-필요시 요청없이도 제공할 수 있음</li> <li>-정보제공에 대한 응답여부는 퍼블리케이션에 의해 결정</li> </ul>	<ul style="list-style-type: none"> <li>-요청 또는 요청이 없더라도 제공정책에 따라 일방적 정보제공이 가능</li> <li>-정보요청 없이 제공시에는 요청에 대한 응답 역시 생략</li> <li>-정보제공에 대한 응답여부는 퍼블리케이션에 의해 결정</li> </ul>

## 4. 기본교통정보교환 기술기준 IV

### 4.1 개요

- 「기본교통정보교환 기술기준IV」은 공공기관에서 무선통신 기술을 이용하여 교통정보를 수집 및 제공할 때 필요한 정보형식을 정의한 것으로, 교통정보 수집 및 제공의 호환성과 연계성을 제고하기 위해 제정됨
  - 제정 2008년 1월 28일, 국토해양부 고시 제2008-45호
  - 최종개정 2016년 4월 15일, 국토교통부 고시 제2016-208호 (일부 개정)

### 4.2 기술기준 구성 및 적용범위

- 본 기술기준은 국가통합교통체계효율화법 제77조 규정에 의한 지능형교통체계 사업에서 무선통신기술을 적용하여 차량의 위치파악 및 구간통행속도 등의 교통정보를 수집하거나, 노면에 설치된 통신장치를 이용하여 교통정보를 제공하는 경우에 적용함
- 또한 국가통합교통체계효율화법 제77조 규정에 의해 시행한 교통체계지능화사업에서 발생하는 교통정보를 활용하거나 이를 이용하여 장비를 제조하는 사업을 영위하는 자를 대상으로 함

### 4.3 정보의 구성

- 기술기준에서 교환대상으로 정의한 기본교통정보는 차량에 장착된 장치(차량장치)에서 노변장치로 송신하는 수집정보와 노변장치를 통해 차량장치로 송신하는 제공정보로 구성되며, 기본정보는 다음과 같음
  - 프로브정보, 교통소통정보, 교통통제정보, 돌발상황정보, 도로상태정보, 기상정보, 위치참조정보

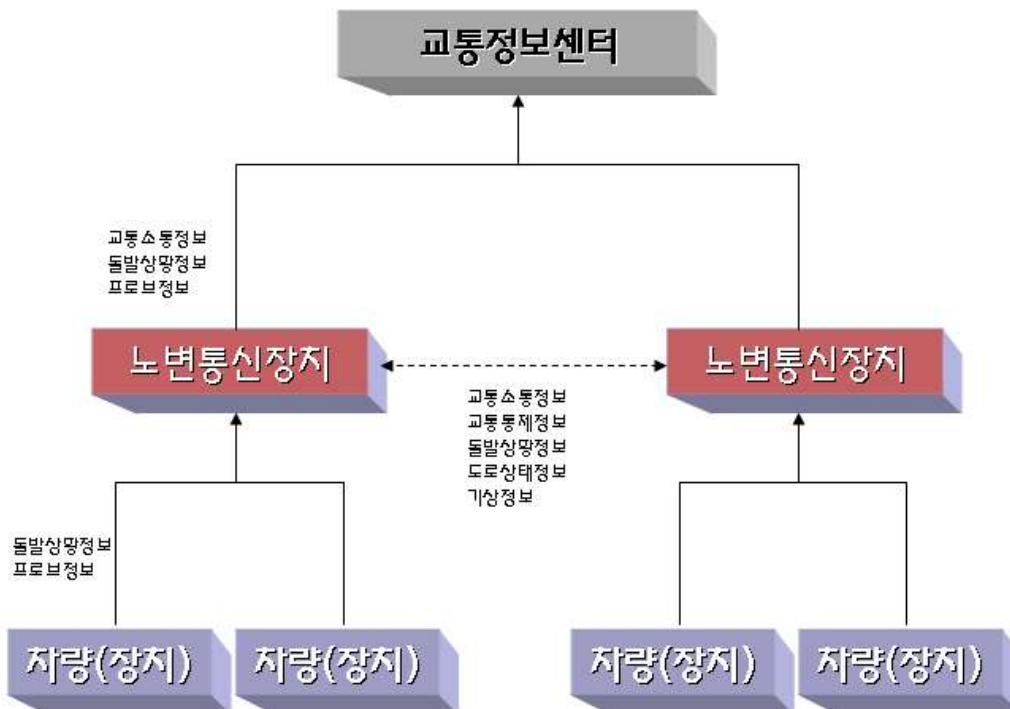
- 공공기관에서 대국민 서비스로 교통정보를 제공하거나 센터의 효율적인 운영을 위해 정보를 교환하는 경우에 사용하는 교통정보는 기본적으로 6개 정보로 구성되며 ID, 교환주기는 아래 표와 같음

<표 13> 기본교통정보교환 기술기준 IV의 정보항목 구성

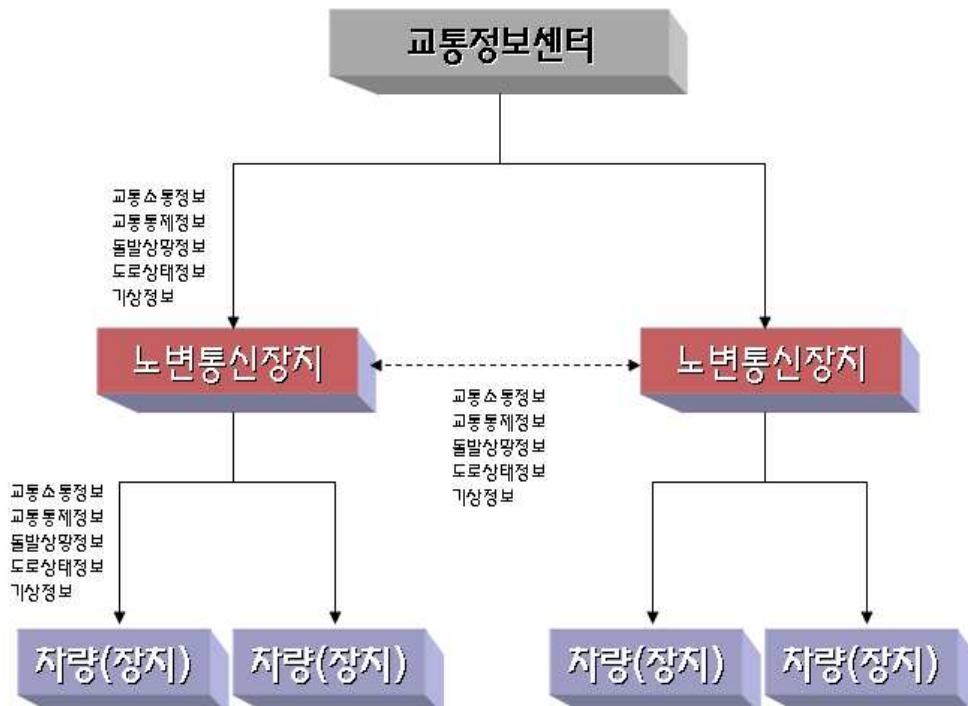
ID	정보명	아키텍처 정보명	정보주기
401	교통소통정보	속도, 교통량, 밀도, 통행시간, 대기길이, 점유율, 지체도, 구간예측평균속도, 구간예측통과시간, 구간속도추이	상시/요청시
402	교통통제정보	위치, 통제유형, 대상, 시간	상황발생시/요청시 /필요시
403	돌발상황정보	위치, 시각, 돌발상황 유형, 돌발상황상태, 정보갱신상태	상황발생시/요청시 /필요시
404	도로상태정보	노면상태, 강우/강설수위	상황발생시/요청시 /필요시
405	기상정보	기온, 날씨, 확률, 가시거리, 풍속, 풍향, 습도, 기압, 일출·일몰시간	요청시/필요시
406	프로브정보	차량종류, 검지시간, 검지위치	상시/발생시

#### 4.4 정보 연계체계

- 교통정보센터는 수집된 정보를 노변통신장치를 통해 차량장치에 제공할 수 있으며, 기본정보는 다음의 5개로 아래와 같은 연계체계를 가짐
  - 교통소통정보, 교통통제정보, 돌발상황정보, 도로상태정보, 기상정보
- 또한 노변통신장치는 정보를 수집하여 교통정보센터에 다음 3개의 정보를 제공할 수 있으며, 교통소통정보의 경우 노변통신장치의 특성 및 시스템 구축비용 등을 고려하여 생략할 수 있음
  - 교통소통정보, 돌발상황정보, 프로브정보
- 그리고 차량장치에서도 돌발상황정보와 프로브정보를 노변통신장치로 제공할 수 있음



〈그림 16〉 기본교통정보교환 기술기준 IV의 상향식 교통정보 연계도



〈그림 17〉 기본교통정보교환 기술기준 IV의 하향식 교통정보 연계도

## 4.5 교통정보세항목 구현방법

### 4.5.1 교통소통정보

#### 401 교통소통정보

본 정보항목은 도로의 교통소통정보를 규약함

정보명	교통소통정보
아키텍처상 정보명	교통정보, 고속도로교통정보, 도시부 간선도로교통정보, 국도/지방도 교통정보
정보세부항목	속도, 교통량, 밀도, 통행시간, 대기길이, 점유율
설명	도로의 교통소통상황을 나타내는 정보

```
CURLINKSTROADSIDE DEFINITIONS AUTOMATIC TAGS:=BEGIN
CurrentlinkStateRoadSide:=SEQUENCE
{
    trmt-MessageCreativeGeneralizedTime VisibleString.
    link-LinkidNumber UTF8String (SIZE(1..40)).
    link-SpeedRate INTEGER (0..300).
    link-VolumeRate INTEGER (0..100000) OPTIONAL.
    link-DensityRate INTEGER (0..2000) OPTIONAL.
    link-TravelTimeQuantity INTEGER (0..10800) OPTIONAL.
    link-DelayQuantity INTEGER (0..12000) OPTIONAL.
    tfdt-VehicleQueueLengthQuantity INTEGER (0..500000) OPTIONAL.
    tfdt-OccupancyPercent INTEGER (0..100) OPTIONAL.
    pctt-PredictionAverageLinkSpeedRate INTEGER (0..300) OPTIONAL.
    pctt-PredictionTraverTimeQuantity INTEGER (0..10800) OPTIONAL.
    pctt-PredictionTimeGeneralizedTime VisibleString OPTIONAL.
    pctt-AdditionalInformationText UTF8String (SIZE(1..256)) OPTIONAL
}
END
```

데이터생성시간
링크식별번호
링크속도
링크교통량
링크밀도
링크통행시간
링크정체
차량대기행렬길이
검지기 점유율
구간예측평균속도
구간예측통과시간
예측시각
소통정보의 추가입력

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTCTime)	날짜, 시각
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
링크속도	주어진 시간동안에 링크를 주행한 차량의 평균 운행속도	km/시
링크교통량	주어진 시간동안에 링크를 통과한 차량	대/시
링크밀도	링크의 단위 킬로미터당 차량수	대/시
링크통행시간	주어진 시간동안에 링크를 주행한 차량의 평균통행시간	초
링크정체	특정 링크를 통행하는 차량의 정체시간	초
차량대기행렬길이	검지기에 의해 측정된 차량들의 평균대기행렬 길이	m
검지기점유율	검지기 점유 백분율 (검지된 시간/총 단위시간×100)	%
구간예측평균속도	해당구간 예측되는 평균속도	km/시
구간예측통과시간	해당구간의 예측되는 차량의 평균통행시간	초
예측시각	예측정보가 적용되는 해당시각	날짜, 시각
소통정보의 추가입력	부가정보의 입력	문자

- 교통소통정보는 정보생성시간, 링크식별번호, 링크속도를 제외한 기타 항목을 선택항목으로 처리하여 정보제공 전략, 목적 등에 적합하게 구성하여 적용할 수 있도록 정의되어 있음
- 링크식별번호는 위치를 나타내기 위한 항목으로 「지능형교통체계 표준노드링크 구축·관리지침」(국토교통부 고시 제2015-755호(전부개정))의 노드·링크 ID 부여방법에 의해 정해진 링크ID를 사용함
- 링크속도는 링크식별번호에 해당하는 링크를 통과하는 평균 속도로써 일정시간 동안 주행한 모든 차량의 평균속도인 시간평균속도를 사용하고 입력 단위는 km/h로 함
- 링크교통량은 주어진 시간동안에 링크를 통과한 차량대수로써 입력단위는 대/시이고, 링크밀도는 단위 킬로미터 당 차량대수로써 입력단위는 대/시임
- 링크통행시간은 주어진 시간동안에 링크를 주행한 차량의 평균통행시간으로써 입력단위는 초이고, 링크지체는 특정링크를 통행하는 차량의 지체시간으로써 입력단위는 초를 사용함
- 차량대기행렬길이와 검지기 점유율은 검지기에 의해 측정된 값으로 차량대기행렬길이는 미터, 점유율은 백분율을 사용함

#### 4.5.2 교통통제정보

##### 402 교통통제정보

본 정보항목은 교통통제정보의 정보 세항목을 규약함

정보명	교통통제정보
아키텍처상 정보명	교통통제정보, 도로정보
정보세부항목	위치, 통제유형, 대상, 시간
설명	미리 계획된 도로의 공사, 행사 등 교통의 통제상황을 나타내는 이벤트 정보

```

EVENTIDENTITYROADSIDE DEFINITIONS AUTOMATIC TAGS ::= BEGIN
EventIdentityRoadSide ::= SEQUENCE
{
    tmt-MessageCreativeGeneralizedTime VisibleString,
    evnt-LocationNodeIdNumber SEQUENCE OF UTF8String,
    evnt-LocationLinkIdNumber SEQUENCE OF UTF8String,
    evnt-LocationTypeCode ENUMERATED OPTIONAL,
    evnt-LaneBlockedOrClosedTypeCode ENUMERATED OPTIONAL,
    evnt-LanesBlockedOrClosedNumber INTEGER (0..255) OPTIONAL,
    evnt-LocationBeginNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    evnt-LocationLinearDistanceOffsetBeginNumber INTEGER (0..99999) OPTIONAL,
    evnt-LocationEndNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    evnt-LocationLinearDistanceOffsetEndNumber INTEGER (0..99999) OPTIONAL,
    evnt-LocationAdditionalInfoText UTF8String (SIZE(1..256)) OPTIONAL,
    evnt-DescriptionTypeEventCode ENUMERATED OPTIONAL,
    evnt-DescriptionTypeSpecialEventCode ENUMERATED OPTIONAL,
    evnt-DescriptionTypeEventOtherText UTF8String (SIZE(1..256)) OPTIONAL,
    evnt-ResponseAlternateRouteText UTF8String (SIZE(1..1024)),
    evnt-LanesShouldersBlockedCode ENUMERATED OPTIONAL,
    evnt-TimelinestarGeneralizedTime VisibleString OPTIONAL,
    evnt-TimelineEndGeneralizedTime VisibleString OPTIONAL,
    evnt-TimelineDurationTime INTEGER (0..4294967295) OPTIONAL
}
END

```

데이터생성시간
이벤트위치노드
이벤트위치링크
이벤트위치유형
차단차로유형
차단차로수
이벤트시작노드
위치선형거리옵셋 시점
이벤트종점노드
위치선형거리옵셋 종점
이벤트위치부연설명
이벤트유형구분
특별이벤트유형설명
이벤트부연설명
대안경로안내
차선길어깨차단코드
이벤트개시시각
이벤트종료시각
이벤트지속표준시

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTC Time)	날짜, 시각
이벤트위치 노드	도로 이벤트가 해당되는 모든 노드번호	-
이벤트위치 링크	도로 이벤트가 해당되는 모든 링크 번호	-
이벤트위치유형	이벤트 통제위치의 유형	-
차단차로유형	이벤트로 인한 차로 차단의 유형	-
차단차로수	도로 이벤트로 인해 차단/통제된 차로수	수
이벤트시작노드	도로 혹은 링크 중간에서 이벤트의 시작특정지점을 표시할 경우 진행방향의 바로 전 노드	번호
위치선형거리옵셋 시점	이벤트 시작노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	m

명칭	설명	단위
이벤트종점노드	도로 혹은 링크 중간에서 이벤트의 종료특정지점을 표시할 경우 진행방향의 바로 다음 노드	번호
위치선형거리옵셋 종점	이벤트 종점노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	m
이벤트위치 부연설명	도로 이벤트의 위치에 대한 부연설명(문자)	문자
이벤트유형구분	각종 도로 이벤트의 유형을 구분할 수 있는 코드	-
특별이벤트유형설명	도로 이벤트의 유형을 표현하기 위한 코드	-
이벤트 부연설명	이벤트에 대한 기타사항, 부연설명	문자
대안경로안내	통제된 도로에 대해 효율적으로 운영할 수 있는 대안경로의 안내	문자
차선길어깨차단코드	도로 이벤트로 인해 차단된 길어깨를 나타내는 코드	-
이벤트개시시각	특별한 도로 이벤트의 실제적인 개시 시각	날짜, 시각
이벤트종료시각	도로 이벤트의 실제적인 종료 시간	날짜, 시각
이벤트지속표준시	도로 이벤트의 실제적인 진행 시간	분

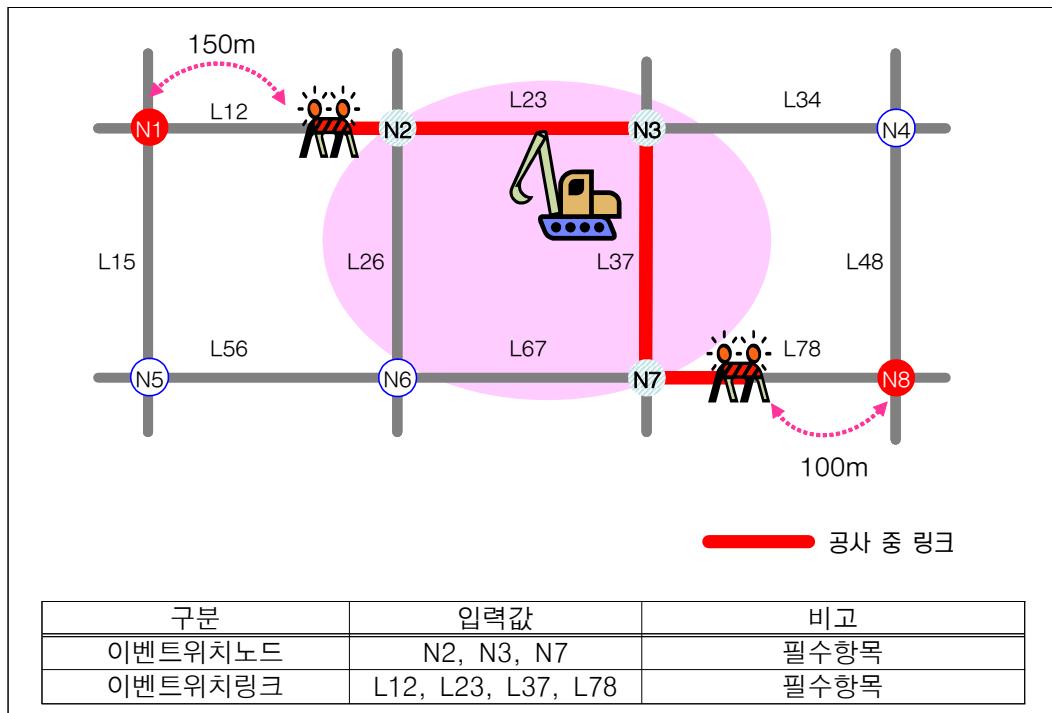
- 교통통제정보는 크게 통제방법 또는 유형, 이벤트발생위치, 시간정보로 구성됨
- 먼저 통제유형을 위한 항목 및 코드값은 다음 표와 같으며, 모두 선택사항으로 정의되어 운영, 또는 정보제공 상 필요한 항목을 선택하여 사용할 수 있음

#### 〈표 14〉 기본교통정보교환 기술기준 IV의 교통통제정보

통제유형구분	통제코드값
이벤트유형구분	없음(0), 기타(1), 공사(2), 도로폐쇄(3)
특별이벤트유형설명	없음(0), 기타(1), 퍼레이드(2), 스포츠이벤트(3), 시위 및 집회(4), 축제(5), 예술공연(6), 수행행렬(7)
계획된 도로차단 유형	없음(0), 기타(1), 장기공사(2), 단기공사(3), 유지보수(4), 긴급유지보수(5)
차선길어깨차단코드	없음(0), 기타(1), 우측길어깨(2), 좌측길어깨(3), 양쪽길어깨(4)

- 기술기준에서 정보의 위치는 표준노드링크 ID로 표현되며, 교통통제 위치의 정확한 표현을 위해서 교통통제정보에서는 표준노드링크ID를 이용한 다양한 이벤트 발생위치표현항목이 포함되어 있음
  - 이벤트위치노드 : 도로이벤트가 발생한 모든 표준노드ID 입력
  - 이벤트위치링크 : 도로이벤트가 발생한 모든 표준링크ID 입력

- 다음 그림은 위치항목 값 입력방법의 예임



<그림 18> 교통통제발생 예시

- 교통통제의 시간정보 제공을 위한 항목은 다음 표와 같이 활용할 수 있음

<표 15> 기본교통정보교환 기술기준 IV의 교통통제 시간정보항목

구분	단위	설명	비고
이벤트개시시각	YYYYYMMDDhhmmss	이벤트의 실제적인 개시시각	선택항목
이벤트종료시각	YYYYYMMDDhhmmss	이벤트의 실제적인 종료시각	선택항목
이벤트지속표준시간	분	이벤트 지속시간으로 이벤트종료시각에서 이벤트개시시각을 뺀 값	선택항목

- 교통통제정보는 정보를 구별할 수 있도록 고유식별자(ID)를 가져야 하고 이는 정보 발생 시마다 일정한 간격(1)으로 증가하여 관리가 가능하도록 하며, 이벤트식별번호(evnt-IdentificationNumber)에 입력하여 전송함
- 교통통제정보에서 코드값 표현방식을 설명하기 위해 차선길어깨차단코드의 경우를 예로 들면, 전송 코드의 입력형식은 BIT STRING 형식으로 다음과 같이 표현할 수 있음

- BIT String은 각 자리를 1또는 0으로 표현하여 해당 자리수가 의미하는 값에 대한 참/거짓으로 표현하여 복합적인 코드를 효과적으로 표현하기 위한 방식
- 정의에서 각 값이 의미하는 것은 bit string의 인덱스를 뜻하며 (0)은 0번 인덱스로 맨 처음자리를 뜻함
- 각 자리수 별로 해당하는 경우 1, 해당하지 않는 경우 0을 설정하여 값을 전송
- 각각의 자리가 의미하는 바에 따라 BIT STRING을 설정하면 다음과 같음

	차단 없음	기타	우측길어깨	좌측길어깨	양쪽길어깨
	Index(0)	Index(1)	Index(2)	Index(3)	Index(4)
차단없는 경우	1	0	0	0	0
기타(본선차단/길어깨통행)	0	1	0	0	0
우측 길어깨만 차단	0	0	1	0	0
좌측 길어깨만 차단	0	0	0	1	0
양쪽 길어깨 차단	0	0	0	0	1

### 4.5.3 돌발상황정보

#### 403 돌발상황정보

본 정보항목은 돌발상황의 처리사항에 대해 규약함

정보명	돌발상황정보
아키텍처상 정보명	돌발상황정보, 돌발상황발생정보, 돌발상황처리요청, 기본정보
정보세부항목	위치, 시각, 돌발상황 유형, 돌발상황상태, 정보갱신상태
설명	예상하지 못한 도로 상황을 알려주는 정보로서 상시 제공받는 정보가 아닌 상황발생시 제공되는 정보

```
TRINFO INCIDENT CONDITIONS DEFINITIONS AUTOMATIC TAGS ::= BEGIN
TrafficIncidentConditions ::= SEQUENCE {
    trnt-MessageCreativeGeneralizedTime VisibleString,
    node-NodeIdNumber UTF8String (SIZE(1..40)),
    link-LinkIdNumber UTF8String (SIZE(1..40)),
    orgn-ContactOrganizationNameText UTF8String (SIZE(1..128)) OPTIONAL,
    evnt-DescriptionTypeIncidentCode BIT STRING OPTIONAL,
    evnt-DescriptionTypeIncidentOther UTF8String (SIZE(1..256)) OPTIONAL,
    evnt-LocationBeginNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    evnt-LocationLinearDistanceOffsetBeginNumber INTEGER (0..99999) OPTIONAL,
    evnt-LocationEndNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    evnt-LocationLinearDistanceOffsetEndNumber INTEGER (0..99999) OPTIONAL,
    evnt-UpdateGeneralizedTime VisibleString OPTIONAL,
    evnt-IncidentStatusCode BIT STRING OPTIONAL,
    evnt-IncidentStatusOther UTF8String (SIZE(1..256)) OPTIONAL,
    evnt-LanesBlockedOrClosedNumber INTEGER (0..255) OPTIONAL,
    evnt-ResponseAlternateRouteText UTF8String (SIZE(1..1024)),
    evnt-UpdateTypeCode BIT STRING OPTIONAL,
    evnt-UpdateTypeOtherText UTF8String (SIZE(1..256)) OPTIONAL
}
END
```

데이터생성시간
이벤트위치노드
이벤트위치링크
연락기관명칭
돌발상황의 유형설명
돌발상황유형의 부연설명
이벤트시작노드
위치선택거리옵션 시점
이벤트종점노드
위치선택거리옵션 종점
이벤트갱신시각
돌발상황상태
돌발상황상태의 부연설명
차단차로수
대안경로안내
갱신유형코드
갱신유형의 부연설명

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTCTime)	날짜, 시각
노드식별번호	네트워크 내 노드에 대한 고유식별번호	번호
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
연락기관명칭	지역 내의 특정 기관의 명칭. 상호관련된 기관이 있을 경우 최상위 기관을 명시	문자
돌발상황의 유형설명	돌발상황에 대한 유형을 표현하기 위한 코드	-
돌발상황유형의 부연설명	돌발상황의 유형에 대한 부연설명	문자
이벤트시작노드	도로 혹은 링크 중간에서 이벤트의 시작특정지점을 표시할 경우 진행방향의 바로 전 노드	번호
위치선형거리옵셋 시점	이벤트 시작노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	m
이벤트종점노드	도로 혹은 링크 중간에서 이벤트의 종료특정지점을 표시할 경우 진행방향의 바로 다음 노드	번호
위치선형거리옵셋 종점	이벤트 종점노드에서 거리로써 측정된 도로 이벤트 발생구간 시점의 정확한 위치	m
돌발상황갱신시각	특별한 도로 이벤트에 대하여 최근 변경시 갱신되어진 시각	날짜, 시각
돌발상황상태	돌발상황 상태를 나타내는 코드	-
돌발상황상태의 부연설명	돌발상황 상태의 부연설명	문자
차단차로수	도로 이벤트로 인해 차단/통제된 차로수	수
대안경로안내	통제된 도로에 대해 효율적으로 운영할 수 있는 대안경로의 안내	문자
갱신유형코드	특별한 도로 이벤트에 대해 정보의 갱신 유형을 나타내는 코드	-
갱신유형의 부연설명	정보의 갱신유형의 부연설명	문자

- 돌발상황정보는 돌발상황의 발생위치, 돌발상황의 유형설명과 이에 대한 부연설명으로 구성되어 있으며, 프로브차량에서 돌발상황정보를 교통정보센터로 전송(제보)하고자 하는 경우 이를 준용할 수 있음
- 갱신유형코드를 적용하여 돌발상황 연장정보 제공 시, 보조정보로 사용함
- 돌발상황 진행 상태에 대한 정보를 제공하기 위해 갱신정보를 전송하며, 이는 갱신유형코드와 갱신유형의 부연설명으로 표현함

<표 16> 기본교통정보교환 기술기준 IV의 돌발상황발생정보 시간항목값 입력(예)

돌발상황상태	갱신유형코드 값
사고감지	신규(0)
처리 및 복구중	업데이트(1)
완료	삭제(2)

- 노드식별번호와 링크식별번호는 돌발상황이 발생된 구간을 표현하기 위한 항목으로 표준노드·링크ID를 사용함
    - 데이터유형은 UTF8String으로 2개 이상의 노드 또는 링크에 걸쳐 발생되었을 경우는 구분자를 사용하여 데이터 입력 가능
    - 돌발정보발생위치를 정확하게 표시하기 위해서는 옵셋거리 활용 가능
  - 연락기관 명칭은 상호 관련된 기관을 명시하기 위한 항목으로 돌발상황의 접수 또는 관할기관 중 최상위기관을 입력함
  - 돌발상황은 발생에서 완료까지 생성주기(Life-Cycle)를 가지며, 하나의 돌발상황마다 고유식별자(ID)를 부여하여 상황 발생 시마다 일정한 간격(1)으로 증가시켜 관리함
  - 돌발상황에 관한 정보에는 돌발상황발생정보와 돌발상황정보가 있으며, 각각 datexPublish-Serial-nbr에 돌발상황 고유식별자(ID)를 입력하여 전송함
  - 그리고 돌발상황정보는 신규/수정/삭제/완료 등 상황에 대한 상태정보를 개선유형코드(Event-UpdateTypeCode)에 입력하여 전송함
- 예) 돌발상황 고유식별자(ID)가 N인 돌발상황발생정보 및 돌발상황정보



<그림 19> 기본교통정보교환 기술기준 IV의 돌발상황 생성주기 관리 (예)

#### 4.5.4 도로상태정보

##### 404 도로상태정보

본 정보양목은 도로노면의 현재 상태정보를 규약함

정보명	도로상태정보
아키텍처상 정보명	도로정보
정보세부항목	노면상태, 강우/강설수위
설명	기상조건 및 돌발상황에 따른 도로의 상태정보

```
TRINFORDWSFST DEFINITIONS AUTOMATIC TAGS ::= BEGIN
TrafficRoadwaySurfaceStatus ::= SEQUENCE {
    trmt-MessageCreativeGeneralizedTime VisibleString,
    node-NodeIdNumber UTF8String (SIZE(1..40)),
    link-LinkIdNumber UTF8String (SIZE(1..40)),
    link-BeginNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    link-EndNodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    link-LengthQuantity INTEGER (0..160000) OPTIONAL,
    link-LocationAdditionalInfoText UTF8String (SIZE(1..256)) OPTIONAL,
    link-SurfaceConditionCode BIT STRING OPTIONAL,
    link-LandBlockedOrClosedCode BIT STRING OPTIONAL,
    evnt-LanesBlockedOrClosedNumber INTEGER (0..255) OPTIONAL,
    evnt-ResponseAlternateRouteText UTF8String (SIZE(1..1024)),
    evsn-DataSurfacePrecipitationDepth INTEGER (0..3000) OPTIONAL,
    evsn-DataSurfaceTemperatureQuantity INTEGER (0..300) OPTIONAL
}
END
```

데이터생성시간
이벤트위치노드
이벤트위치링크
시점노드식별번호
종점노드식별번호
링크길이
지역추가정보
도로노면상태
도로차단여부
차단차로수
대안경로안내
강우/설량수위
도로표면온도

명칭	설명	단위
정보생성시각	해당정보의 생성시각 (UTC Time)	날짜, 시각
노드식별번호	네트워크 내 노드에 대한 고유식별번호	번호
링크식별번호	네트워크 내 링크에 대한 고유식별번호	번호
시점노드식별번호	링크 시점에 해당되는 노드의 고유식별번호	번호
종점노드식별번호	링크 종점에 해당되는 노드의 고유식별번호	번호
링크길이	시점노드에서 종점노드까지의 링크 길이	m
지역추가정보	이용자가 명확히 파악할 수 있는 ASCII코드로 된 지역정보	문자
도로노면상태	링크의 현재 도로노면 상태	-
도로차단여부	도로 차단여부	-
차단차로수	도로 이벤트로 인해 차단/통제된 차로수	수
대안경로안내	통제된 도로에 대해 효율적으로 운영할 수 있는 대안경로의 안내	문자
강우/설량수위	환경감지기에서 노면의 강우량 수위를 나타내는 숫자적인 색인	mm
도로표면온도	환경감지기에서 섭씨로 측정된 도로의 표면 온도	°C

- 도로상태정보의 위치를 나타내기 위한 세항목으로는 노드식별번호와 링크식별번호가 있으며, 돌발상황정보와 같이 2개 이상의 노드 또는 링크에 대한 도로상태 정보를 전송하기 위해서는 구분자를 사용하여 노드와 링크값을 입력할 수 있음
- 도로노면상태는 기후 또는 돌발상황, 외부환경변화 등에 의해 변화된 노면상태정보를 제공하기 위한 항목으로, 중복선택이 가능하며 “노면습윤상태”, “적설”의 경우는 강우/설량수위로 정보를 추가적으로 제공할 수 있음
- 도로차단여부는 노면상태에 따른 도로의 차단여부를 나타내는 것임

<표 17> 기본교통정보교환 기술기준 IV의 도로 및 차로차단 경우의 수

구분	도로 차단여부
도로차단안함/차로차단안함	차단안됨(3)
도로차단안함/차로일부차단	차단안됨(3)
도로차단/전차로 차단	차단됨(4)

#### 4.5.5 기상정보

##### 405 기상정보

본 정보항목은 해당구간의 기상상황 정보를 규약함

정보명	기상정보
아키텍처상 정보명	기상정보
정보세부항목	기온, 날씨, 확률, 가시거리, 풍속, 풍향, 습도, 기압, 일출/일몰시간
설명	일반적인 기상상황을 나타내는 기상정보

WETHROADSIDE DEFINITIONS AUTOMATIC TAGS ::= BEGIN	데이터생성시간
WeatherInformationRoadSide ::= SEQUENCE	기상예측유무
{	기온
wthr-CreativeGeneralizedTimeVisibleString.	최고기온
wthr-ForecastOrActualCodeENUMERATED,	최저기온
wthr-TemperatureQuantityINTEGER (-128..127) OPTIONAL,	기상특정상태
wthr-HighTemperatureQuantityINTEGER (-128..127) OPTIONAL,	기상특정상태의 부연설명
wthr-LowTemperatureQuantityINTEGER (-128..127) OPTIONAL,	기상예보확률
wthr-SpecialConditionsBIT STRING(256)) OPTIONAL,	가시거리
wthr-SpecialConditionsOther UTF8String (SIZE(1..256)) OPTIONAL,	풍속
wthr-ProbabilityPercent INTEGER (0..100),	풍향
wthr-VisibilityQuantity INTEGER (0..255) OPTIONAL,	습도
wthr-WindSpeedQuantityINTEGER (0..255) OPTIONAL,	기압
wthr-WindDirectionCodeENUMERATED,	일출시간
wthr-HumidityQuantity INTEGER (0..100) OPTIONAL,	일몰시간
wthr-PressureQuantityINTEGER (0..100) OPTIONAL,	
wthr-SunriseTimeGeneralizedTimeVisibleString OPTIONAL,	
wthr-SunsetTimeGeneralizedTimeVisibleString OPTIONAL	

명칭	설명	단위
기상정보제공위치	기상정보가 제공되는 위치의 명칭	문자
기상정보생성시각	기상정보가 생성된 시각	날짜, 시각
기상예측유무	기상의 유형으로 실제와 예측으로 구분됨	-
기온	온도	°C
최고기온	최고기온	°C
최저기온	최저기온	°C
기상예보확률	기상예측에 대한 확률적 평가	%
기상특정상태	날씨 정보의 특정상태로서 적란운에 의해서 생성되는 비의 대기 조건 또는 태풍 등과 같은 기상현상을 설명함	-
기상특정상태의 부연설명	날씨정보의 부연설명	문자
가시거리	가시거리로서 m로 나타냄	m
풍속	바람의 속도	m/s
풍향	바람의 방향	-
습도	습도	%
기압	기압	mb
일출시간	일출시간	-
일몰시간	일몰시간	-

- 기상정보에서 기상예측유무는 필수로 제공하여야 하며 이외의 메시지는 선택적으로 제공함
- 현재의 기상정보와 예측된 기상정보 모두 제공이 가능하며 정보 요청 시에만 기상청에서 제공받을 수 있음
- 기상예측유무 및 기상특정상태를 나타내는 코드는 다음 표와 같음

<표 18> 기상예측유무 및 기상특정상태 코드

구분	설명	통제코드값
기상예측유무	기상정보가 관측된 정보인지 예측된 정보인지를 구분	현황기상정보(0), 예측기상정보(1)
기상특정상태	기상의 상태에 따른 구분	추가정보필요없음(0), 추가정보필요(1), 눈사태(2), 구름(3), 강추위(4), 폭염(5), 홍수(6), 안개(7), 태풍(8), 오존경보(9), 호우/대설(10), 놀우(11), 자외선경보(12)

- 기온, 최고기온 및 최저기온의 사용단위는 °C, 기상예보확률 및 습도의 사용단위는 %, 가시거리의 사용단위는 m, 풍속의 사용단위는 m/s, 기압의 사용단위는 mb, 일출시간 및 일몰시간의 사용단위는 시각임

#### 4.5.6 프로브정보

##### 406 프로브정보

본 정보항목은 교통소통연장상황을 알려주는 프로브 정보를 규약함

정보명	프로브정보
아키텍처상 정보명	프로브정보, 위치정보
정보세부항목	차량종류, 감지시간, 감지위치
설명	프로브 차량을 이용하여 수집되는 기본교통정보

```

TRAFFICPROBEROADSIDE DEFINITIONS AUTOMATIC TAGS ::= BEGIN
ProbeVehicleDetectionRoadSide ::= SEQUENCE
{
    vpbd-VehicleIdNumberUTF8String (SIZE(1..256)),
    vpbd-VehicleClass ENUMERATED,
    vpbd-VehicleTimeIn GeneralizedTimeVisibleString,
    node-NodeIdNumber UTF8String (SIZE(1..40)) OPTIONAL,
    vpbd-VehicleLocatorLatitude INTEGER (-4294967295 .. -3000000000) OPTIONAL,
    vpbd-VehicleLocatorLongitude INTEGER (3000000000..4294967295) OPTIONAL
}
END

```

차량식별번호
차량분류
차량진입시각
노드식별번호
차량위치위도
차량위치경도

명칭	설명	단위
차량식별번호	개별 프로브 차량에게 배정된 고유한 태그 식별번호	번호
차량분류코드	프로브 차량 유형에 대한 차종 분류 코드	-
차량진입시간	프로브 차량이 특정위치에 감지된 시간 또는 위치정보를 생성한 시각	시각
노드식별번호	네트워크 내 노드에 대한 고유식별번호로 프로브 차량의 위치정보가 제공될 지점번호	번호
차량위치	차량이 위치정보를 제공하는 지점의 경위도 좌표	문자

- 프로브 정보는 주기적으로 수집되는 교통소통정보를 보완·검증·확인하기 위하여 OBU 탑재차량을 투입하여 수집하는 정보를 말함
- 차량식별번호는 개별 프로브 차량에 배정된 고유한 식별번호임
- 차량분류는 프로브 차량 유형에 대한 차종 분류 코드는 다음과 같음

<표 19> 차량분류 코드

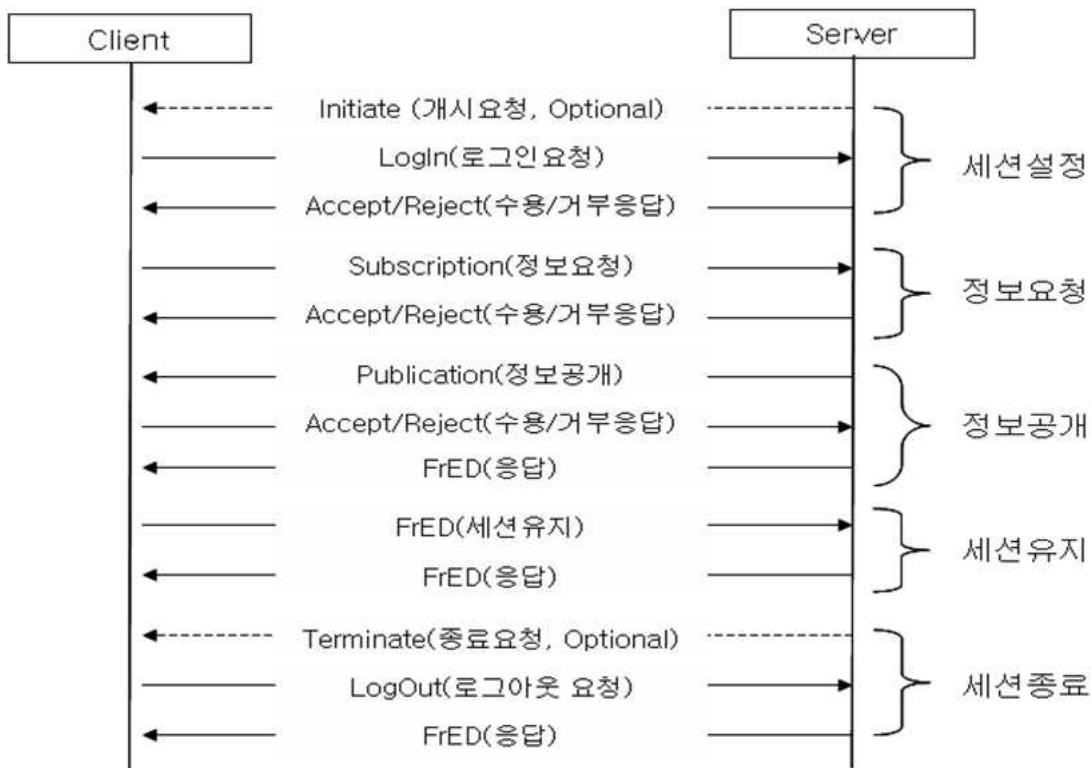
구분	설명	통제코드값
차량분류코드	프로브 차량 유형에 대한 차종 분류	추가정보필요없음(0), 추가정보필요(1), 버스(2), 택시(3), 일반승용차(4), 응급차량(5), 관용차량(6)

- 차량진입시간 및 차량진출시간은 프로브 차량이 도로 상/하류부에 설치된 AVI 판독기에 감지된 시간임

- 차량통행시간은 자동차량인식기에 의해 파악된 프로브 차량의 평균 링크 통행시간으로 사용단위는 초임
- 차량평균속도는 자동차량인식기를 통해 파악한 프로브 차량의 평균운행속도로 사용단위는 Km/시임
- 프로브차량의 위치는 AVI를 이용한 경우에는 AVI가 설치된 지점의 위도와 경도를, GPS를 이용한 경우에는 GPS 위치탐지기가 설치된 위도와 경도를 사용함
  - 시점노드위치, 종점노드위치는 국제표준(IEC61162)의 위치정보 표기법 중 \$GPGGA의 경위도 좌표를 입력함  
예) \$GPGGA, 137809.000, 3732.6627, N, 12701.3549, E, ... 중 경위도 좌표인 3732.6627, N, 12701.3549, E를 입력

## 4.6 통신절차

- 통신절차는 세션설정, 정보요청, 정보공개, 세션유지, 세션종료로 구분됨



<그림 20> 센터-노변장치 간 데이터교환절차



<그림 21> 노변장치-차량장치 간 데이터교환절차

## 5. 대중교통(버스) 정보교환 기술기준

### 5.1 개요

- 버스정보시스템 구축 사업의 개별적 추진으로 인해서 서로 다른 버스정보센터 간 정보연계 시 정보항목별 매칭룰(matching rule)이 필요하게 됨
- 본 기준은 대중교통(버스)에 관한 시스템 구축·운영의 효율성, 정보수집 및 정보교환의 신속성 및 정확성, 센터 간 정보교환 연계성을 확보하고자 함

#### 호환/연동성 확보

지자체 버스정보화 사업간 대중교통정보 연계기반 마련(수평적 정보연계)

교통정보/대중교통정보 연계를 위한 상/하위 센터간 정보 연계기반 마련(수직적 정보연계)

#### 구현방향 제시

사업추진 중 및 계획단계의 공공사업에 정보 수집, 생성, 분석, 가공, 제공 단계별 구현방향 제시

- 본 기술기준은 2005년에 처음으로 제정되었고 2010년과 2014년에 시스템 구축현황과 기술발전에 발맞춰 개정되었으며, 최근 개정 필요성에 대한 검토가 진행 중에 있음 (\*참고: 2016년 재검토기한 연장에 따라 일부 개정됨)
  - 2010년 주요 개정사항
    - 표준적용범위에서 통신장치, 표시장치 등이 제외되면서 센터-센터 및 센터-연계시스템으로 정보연계를 함께 수정
    - 기술기준의 내용이 「버스정보시스템의 기반정보 구축 및 관리요령」과 상충되지 않도록 용어를 통일하고 유효값을 조정
    - 실제 구축 및 운영되는 시스템과 기술발전 현황을 고려하여 정보내용 및 속성을 수정·보완
  - 2014년 주요 개정사항
    - M버스라 불리는 광역급행버스 이용 편의를 위하여 잔여좌석정보를 추가하였으며, 광역급행버스가 운행하지 않는 지역을 고려하여 정보항목을 선택사항으로 규정
    - 잔여좌석정보는 실시간으로 전송되는 버스위치정보와 도착예정정보에 포함하여 전송할 수 있도록 정보항목을 신설

### 5.2 기술기준 구성 및 적용범위

- 자치단체, 공공법인 또는 공공단체 등 공공기관이 구축·운영하는 시스템(이하 “공

공시스템”이라 함) 또는 공공시스템과 연계하는 시스템에 적용함

- 본 기준은 원칙적으로 다수의 정류장을 정차하는 노선버스(여객자동차운수사업법에서 정의하는 노선여객자동차운송사업 중 시외버스운송사업의 고속형 버스 제외)를 대상으로 함

〈표 20〉 대중교통(버스)정보교환 기술기준의 적용범위

항목	세부 항목	설명	적용여부
시스템 구축목적	버스정보시스템	대시민 정보제공을 위한 시스템	○
	버스운행관리시스템	버스운행관리 및 통제를 위한 시스템	○
시 경계 내·외	시내버스	일반버스, 좌석버스	○
		마을버스	✗ <sup>1)</sup>
	시외버스	일반버스, 좌석버스	○
		공항버스, 고속버스	✗ <sup>2)</sup>
정보이용 주체	이용자	대기승객, 차내승객, 운전자	○
	센터 운영자	공공부문 센터 운영자	○
	센터간 연계	공공부문 센터간 정보 연계	○
정보연계 주체	수평적 연계	지자체 센터간, 권역 센터간 정보 연계	○
	수직적 연계	지자체, 권역, 통합교통정보 센터간 정보연계	○

주1) 마을버스는 시내운행 특성상 도시와 도시간 정보연계 대상 아님

주2) 공항버스, 고속버스는 민간부문으로서 공공부문과 연계 구현 사례 없음

### 5.3 정보의 구성

- 대중교통(버스) 정보교환을 위한 기술기준에서 제시하는 연계정보는 정보의 수집 목적에 따라 버스운행관리를 위한 정보와 버스안내정보로 구분됨
- 한편 정보항목은 정보수혜자를 위한 제공정보, 단일 버스정보/운행관리시스템 센터에 수집, 가공, 산출되는 처리정보로 구분됨
  - 버스관리정보 : 운행간격 조정 및 승객보호, 운행지시 등을 위하여 버스위치, 운행상태, 긴급상황, 운행계획, 운행지시에 관한 정보로 구성
  - 버스안내정보 : 이용객의 버스선택 및 승차, 하차편의를 위하여 버스위치, 도착예정, 운행계획으로 구성
  - 제공정보 : 버스도착 및 운행정보 제공을 통한 시민의 대중교통 이용편의성 증진과 운영효율화를 목적으로, 현장에서 버스 소통정보를 수집, 센터에서 분석, 가공,

산출되어 이용자에게 최종적으로 제공되는 정보

- 처리정보 : 버스정보 수집 및 분석, 가공 등 서비스 제공 주체 및 운영감독 주체로서 공공부문(지자체) 버스정보센터 내 운영·처리 정보와 버스운행계획 등 정적·주기적인 편성, 업데이트되는 정보



<그림 22> 대중교통(버스)정보교환 기술기준 정보항목 특성

- 버스정보는 정보교환주기, 인터페이스에 따라 다음과 같이 정보별 ID와 내용을 구분함

<표 21> 대중교통(버스)정보교환 기술기준 정보항목

ID	정보명	정보내용	교환주기	인터페이스	아키텍처상 정보명
201	버스위치 정보	차량ID, 노선ID, 막차 여부, 막차의 최종 도착정류장ID, 이벤트정보-메시지발생시각, 이벤트정보수집노드(zone/구역)ID, 노드진입·진출시각, 노드통행시간, 정주기정보-차량위치정보(GPS), 위치정보수집시각, 정보수집주기, 버스잔여좌석정보	실시간	센터-센터	버스위치 정보
202	도착예정 정보	정류장ID, 노선ID, 차량ID, 도착예정시간(출발정류장ID 및 진출·진입시각, 통과시간), 평균통행속도, 막차정보, 버스잔여좌석정보	실시간	센터-센터	도착예정 정보
203	운행계획 정보	노선기본정보(노선ID, 노선명칭, 기·종점정류장ID), 노선부가정보, 노선운행정보(첫차막차 출발시각, 첨두/비첨두 배차간격), 차량운행횟수, 운행계획정보 갱신시각/갱신내용	변경시	센터-센터	운행계획 정보
204	운행지시 정보	차량ID, 다음정차정류장ID, 차간거리조정, 운행지시정보	필요시	센터-센터	운행조정 정보
205	운행관리 정보	차량ID, 이벤트정보수집노드(zone, 구역)ID, 차량위치정보(GPS), 운행상태정보, 무단결행노선ID, 무단결행발생대수, 노선ID	필요시	센터-센터	운행상태 정보
206	긴급상황 정보	차량ID, 노선ID, 이벤트정보수집노드(zone, 구역)ID, 차량위치정보(GPS), 발생위치(도로명칭, 관련교차로, 돌발상황발생위치설명), 돌발상황발생시각, 돌발상황부연설명, 돌발상황유형, 돌발상황부연설명, 돌발상황긴급정도, 돌발상황긴급정도 부연설명	유고발생시	센터-센터	돌발상황보완정보

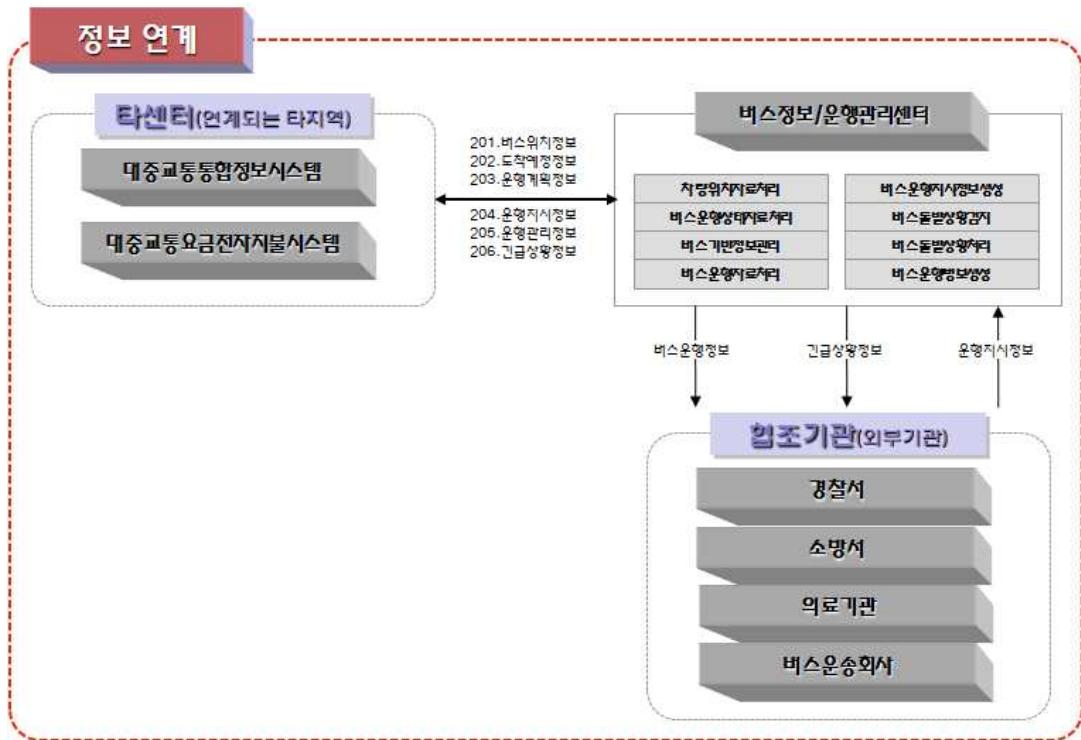
〈표 22〉 대중교통(버스)정보교환 기술기준 정보항목 분석

정보명	실시간정보	제공정보	연계정보	사용주체	비고
버스위치정보	∨	∨	∨	대기승객, 운영자(연계), 운전자(회사)	-
도착예정정보	∨	∨	∨	대기승객, 운영자(연계)	시간 및 위치정보
운행관리정보	∨	-	∨	운영자(연계), 운전자(회사)	상태정보 위반정보
운행계획정보	-	∨	∨	대기승객, 운영자(연계), 운전자(회사)	-
운행지시정보	-	∨(운전자)	∨	운영자, 운전자(회사)	-
긴급상황정보	-	-	∨	대기승객(필요시) 운영자, 운전자(회사)	-

- 정보내용 중 차량ID, 정류장ID, 노선ID 설정과 운행계획 작성은 버스 정보교환의 호환성 확보를 위하여 「버스정보시스템의 기반정보 구축 및 관리요령(국토교통부 고시 제2013-252호(개정))」에서 규정한 사항을 따르도록 함
  - 상기 요령을 기반으로 한 관리시스템이 구축·운영 중일 경우, 관리시스템의 기반정보를 우선하며, 센터간의 ‘203. 운행계획정보’는 교환하지 않아도 되는 것으로 함
- ‘204. 운행지시정보’, ‘205. 운행관리정보’는 정보수집주체와 제공주체가 다를 경우(필요시) 사용하는 것으로 하며, 사용 시에는 본 기술기준에서 정의하는 대로 구현하도록 함

## 5.4 정보연계

- 정보연계란 다수의 지자체를 운행하는 도시 간 광역노선버스의 경우, 해당 자치단체의 측위 및 무선통신체계만으로는 연속적인 위치파악 등 운행관제가 불가능하기 때문에 인접 지자체간 또는 권역/광역버스정보센터 간 정보교환을 기반으로 운행관제 및 정보제공의 연속성을 유지하기 위해 버스운행정보를 시스템 간에 공유하는 것을 말함
- 버스정보는 수집방식 및 가공주체의 상이, 버스정보의 연계/통합 여부에 불구하고 상호 호환되도록 아래 그림과 같이 연계되어야 함



<그림 23> 센터 간 정보연계에 따른 정보흐름

- 버스관리정보 및 버스안내정보를 수집/가공하는 주체가 동 정보의 신뢰도를 확보하는 것을 원칙으로 함
- 버스정보 수집/가공/제공단계의 관리주체가 서로 다른 경우에는 각 단계의 행위 주체가 동 정보의 신뢰도를 확보하며, 정보연계의 경우에는 제공주체가 소관 정보의 신뢰도를 확보해야 함

## 5.5 교통정보세항목 구현방법

### 5.5.1 버스위치정보(201)

#### 1) 개요

- 버스위치정보는 대중교통차량의 실시간 위치정보를 정의하기 위한 정보규약으로서 정류장대기승객 및 버스 내 승객을 위한 실시간 버스위치정보, 센터운영자 및 연계정보 요청 시에 전송하는 센터운영자의 버스관제정보 등으로 정보의 효용성이 매우 높기 때문에 기술기준으로 정보형식 및 내용이 규약되어야 함

201 버스위치정보		본 정보항목은 버스의 실시간 위치(통과)정보를 규약함
정보명	버스위치정보	
아키텍처상 정보명	버스위치정보	
정보세부항목	차량ID, 노선ID, 막차의 최종 도착정류장 ID, 이벤트정보-메시지발생시각, 이벤트정보수집노드(zone/구역)ID, 노드진입·진출시각, 노드통행시간, 정주기정보-차량위치정보(GPS), 위치정보수집시각, 정보수집주기, 버스잔여좌석정보	
설명	버스의 위치정보가 관할지역 이외의 지역에서 수집될 경우 정보가공 및 제공을 위해 타지역 센터로 버스위치정보를 다음과 같이 연계함	
<pre> BusLocationInfo ::= SEQUENCE {     tsfc-PTVehicleIDNumber           UTF8String      (SIZE(1..255)),     busLocationInfoType              CHOICE     {         busLocationEvent            BusLocationEvent,         busLocationPolling          BusLocationPolling     },     tpif-SubRouteIdentityNumber       UTF8String      (SIZE(1..255)),     tsfc-LastPTVehicle               BOOLEAN         OPTIONAL,     tpif-LastPTVehicleBTTIdentifyNumber   UTF8String      (SIZE(1..40)) OPTIONAL,     tsfc-BUSRemainingSeatsNumber     INTEGER         (0..512)    OPTIONAL } BusLocationEvent ::= SEQUENCE {     tsmg-MessageGenerationTime      GeneralizedTime,     tsvh-NodeZoneIDNumber           UTF8String      (SIZE(1..40)),     tsvh-NodeZoneEntryTime          GeneralizedTime OPTIONAL,     tsvh-NodeZoneExitTime           GeneralizedTime OPTIONAL,     tsvh-NodeZoneTripTime           INTEGER         (0..1200)   OPTIONAL } BusLocationPolling ::= SEQUENCE {     tsfc-PTVehicleCoordinate        NMEACoord      OPTIONAL,     tsfc-PTVehicleCollectedTime    GeneralizedTime OPTIONAL,     tsfc-PTVehicleCollectedCycleTime INTEGER      (0..3600)   OPTIONAL } NMEACoord ::= SEQUENCE {     latitude      SEQUENCE {         integerValue   INTEGER,         fractionValue INTEGER,         fracSize       INTEGER      (0..10),         compass        ENUMERATED { north, south }     },     longitude     SEQUENCE {         integerValue   INTEGER,         fractionValue INTEGER,         fracSize       INTEGER      (0..10),         compass        ENUMERATED { east, west }     },     optData        OCTET STRING      OPTIONAL } </pre>		<div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <div style="background-color: #f0f0ff; padding: 5px;">정보의 이름/시각</div> <div style="background-color: #f0f0ff; padding: 5px;">차량ID</div> <div style="background-color: #f0f0ff; padding: 5px;">버스위치정보 구분</div> <div style="background-color: #f0f0ff; padding: 5px;">노선ID</div> <div style="background-color: #f0f0ff; padding: 5px;">막차정보</div> <div style="background-color: #f0f0ff; padding: 5px;">막차의 종점 정류장ID</div> <div style="background-color: #f0f0ff; padding: 5px;">버스잔여좌석정보</div> <div style="background-color: #f0f0ff; padding: 5px;">이벤트정보</div> <div style="background-color: #f0f0ff; padding: 5px;">메시지 발생시각</div> <div style="background-color: #f0f0ff; padding: 5px;">이벤트정보수집노드ID</div> <div style="background-color: #f0f0ff; padding: 5px;">노드진입시각</div> <div style="background-color: #f0f0ff; padding: 5px;">노드진출시각</div> <div style="background-color: #f0f0ff; padding: 5px;">노드통행시간</div> <div style="background-color: #f0f0ff; padding: 5px;">정주기정보</div> <div style="background-color: #f0f0ff; padding: 5px;">차량위치정보(GPS)</div> <div style="background-color: #f0f0ff; padding: 5px;">위치정보수집시각</div> <div style="background-color: #f0f0ff; padding: 5px;">정보수집주기</div> <div style="background-color: #f0f0ff; padding: 5px;">NMEA Coord</div> <div style="background-color: #f0f0ff; padding: 5px;">위도정보</div> <div style="background-color: #f0f0ff; padding: 5px;">경도정보</div> <div style="background-color: #f0f0ff; padding: 5px;">NMEA 좌표 추가정보</div> </div> </div>

명칭	설명	단위
차량ID	대중교통 사업자의 각 차량에 지정된 고유번호	-
노선ID	대중교통 노선의 고유식별자	-
막차정보	현재 진행하는 차량의 막차여부	-
버스잔여좌석정보	광역급행버스에서 사용되는 정보로써 버스잔여좌석정보	명
막차의 종점 정류장ID	해당노선의 막차가 진행하는 방향의 종점(최종정류장)의 고유식별자	-
이벤트 정보	노드영역 중심으로 생성하는 정보	-
메시지 발생시각	해당시스템에 따라 정보가 생성된 시각	-
이벤트정보수집노드ID	대중교통 차량이 정해진 노선을 운행할 때 경유하게 되는 노드영역(교차로, 도로상의 정보수집지점, 정류장 등)의 고유번호	-
노드진입시각	대중교통 차량이 노드영역으로 진입한 시각	시각
노드진출시각	대중교통 차량이 노드영역을 벗어난 시각	시각
노드통행시간	대중교통차량이 노드영역에 머무른 시간, 정차시간	초
정주기 정보	수집주기 마다 생성하는 정보	-
차량위치정보 (GPS)	대중교통 차량의 GPS 위치정보	-
위치정보수집시각	대중교통 차량의 위치를 수집한 시각	시각
정보수집주기	대중교통 차량의 수집주기	초

- 노선은 여객자동차운수사업법 제3조 제1항 제1호의 규정에 따라 기·종점 및 승·하차지점을 정하여 정기적으로 여객운송용 자동차를 운행하기 위한 구간으로 노선ID는 버스정보시스템의 기반정보 구축 및 관리요령에서 규정한 노선ID를 사용하도록 함
- 센터간 대중교통차량 위치정보를 GPS 좌표값으로 적용할 경우에는 NMEA 좌표정보값을 기준으로 교환함
  - NMEA Coord : 표준 NMEA 좌표 정보

명칭	설명	단위
위도정보	NMEA좌표 정보 중 위도 정보	-
	NMEA 좌표 정보 중 소수 점 이상 정수 정보	-
	NMEA 좌표 정보 중 소수 점 이하 정보	-
	소수 점 이하 정보의 총 자릿수	-
	남/북 방향 정보	-
경도 정보	NMEA좌표 정보 중 경도 정보	-
	NMEA 좌표 정보 중 소수 점 이상 정수 정보	-
	NMEA 좌표 정보 중 소수 점 이하 정보	-
	소수 점 이하 정보의 총 자릿수	-
	동/서 방향 정보	-
NMEA좌표 추가정보	NMEA좌표 정보 제공 시 선택적 추가정보	-

- 한편 구현되는 무선통신 및 측위시스템 방식과 무관하게 준수할 수 있어야 하므로 기술기준에서는 이벤트정보와 정주기정보로 구분하여 협역무선통신 및 측위방식(비콘, DSRC, 무선랜 등)에 적용되는 이벤트정보를 강제항목으로 하여 광역무선통신 및 측위방식(GPS + 무선데이터, CDMA, TRS 등)으로 구현가능한 정주기 정보를 권고항목으로 정의하고 있음

## 2) 정보생성

- 버스위치정보는 적용되는 무선통신방식 및 위치추적방식에 따라 구현방법상의 차이를 가짐
- 협역무선통신 및 측위방식의 경우에는 비콘, DSRC, 무선랜 등 노변통신장치의 통신영역(communication zone)에 대중교통차량이 진입, 진출하는 시각을 검지하여 특정지점에서 대중교통차량의 도착, 출발 등 통과시각을 산출하는 방식이기 때문에 이벤트정보(event data) 수집이 가능한 방식임
- 반면 광역무선통신 및 측위방식의 경우에는 무선데이터, CDMA, TRS 등 비교적 통신반경이 1.5~수km에 달하는 기지국, 중계기의 통신영역 내부를 운행하는 대중교통차량의 위치정보를 GPS 측위좌표를 이용하여 특정 지점좌표에 대한 반경(30~50m 등)의 설정된 영역으로 도착, 출발 여부를 검지하는 방식으로 구현됨
- 따라서 광역무선통신 방식은 이벤트정보 수집은 물론, 대중교통차량 내부에 장착된 단말기 주기 설정값에 의한 타이머전송 방식의 정주기정보 수집이 가능한 방식임
- 기술기준에서 제시하고 있는 버스위치정보형식은 사업에 적용된 무선통신방식의 협역, 광역 특성과 독립적으로 정보형식을 규약하기 위해서 이벤트정보를 강제항목, 정주기 정보를 권고항목으로 규정하고 있으며 각 정보의 세부적인 규약내용은 다음과 같음

### 3) 상세설명

- 버스위치정보는 정보정의 시작(BUSLOCATIONINFO DEFINITION AUTOMATIC TAGS ::=BEGIN)으로부터 크게 이벤트정보 부분과 정주기정보 부분으로 구분되어 있음
- 이벤트정보의 내용은 다음과 같음

<표 23> 버스위치정보/ 이벤트정보 정의내용

BusLocationEvent ::= SEQUENCE {	①	{
tsmg—MessageGenerationTime   GeneralizedTime,	②	
tsvh—NodeZoneIDNumber       UTF8String   (SIZE(1..40)),	③	
tsvh—NodeZoneEntryTime      GeneralizedTime                   OPTIONAL,	④	
tsvh—NodeZoneExitTime       GeneralizedTime                   OPTIONAL,	⑤	
tsvh—NodeZoneTripTime      INTEGER       (0..1200)   OPTIONAL		
}		}
① : 버스위치정보정의 시작(특정지점 통과정보, 이벤트정보)	①	
② : 이벤트정보 메시지생성시각, 즉 대중교통차량의 이벤트 발생시각	②	
③ : 노드영역(통신영역, 설정된 좌표로부터의 반경내부)에 진입한 시각	③	
④ : 노드영역을 진출한 시각	④	
⑤ : ③-④로 계산되는 노드영역에 잔류한 시간, 즉 통행시간, 정차시간	⑤	

- 정주기 정보의 내용은 다음과 같음

<표 24> 버스위치정보/ 정주기정보 정의내용

BusLocationPolling ::= SEQUENCE {	①	{
tsfc—PTVehicleCoordinate     NMEACoord                   OPTIONAL,	②	
tsfc—PTVehicleCollectedTime   GeneralizedTime           OPTIONAL,	③	
tsfc—PTVehicleCollectedCycleTime   INTEGER       (0..3600)   OPTIONAL		
}		}
① : 버스위치정보정의 시작(특정시점 위치정보, 정주기정보)	①	
② : 대중교통차량의 GPS 위치정보	②	
③ : 위치정보 수집시각	③	
④ : 정보 수집주기		

## 5.5.2 도착예정정보(202)

### 1) 개요

- 도착예정정보는 정보제공대상인 대중교통차량이 경유하는 정보제공대상 정류장에 도착하는데 소요되는 시간정보와 대중교통차량의 현재 위치정보를 규약함
- 또한 도착예정정보는 정류장대기승객과 연계정보를 요청하는 센터운영자 측면에서 정보효용성이 높기 때문에 기술기준으로 정보형식 및 내용을 규약할 필요가 있음
- 도착예정정보는 무선통신 및 측위시스템 방식과는 무관하게 적용할 수 있도록 하고 도착예정정보 산출과정에 도입되는 변수를 생략함으로써 시스템의 자율성을 보장하는 링크통행시간과 경유하는 교차로, 정류장에서의 정지시간 및 정차시간의 합계로 표현되는 도착예정시간과 현재 버스의 위치정보로 정의하고 있음

### 202 도착예정정보

본 정보항목은 버스의 실시간 도착예정시간 및 통과지점 정보를 규약함

정보명	도착예정정보
아키텍처상 정보명	도착예정정보
정보세부항목	정류장ID, 노선ID, 차량ID, 도착예정시간(출발정류장ID 및 진출·진입시각, 통과시간), 평균통행속도, 막차정보, 버스잔여좌석정보
설명	정류장 단말기, 차내 안내단말기, 개인단말기에 직접 제공 또는 인접센터 및 광역센터와 연계를 통한 중계제공 또는 재가공(긴급상황 등을 반영) 하여 제공

ArrivalPredictionTimeInfo ::= SEQUENCE {		정보의 이름/시작	
tsmg	MessageGenerationTime	GeneralizedTime,	메세지발생시각
tpif	BITIdentifyNumber	UTF8String (SIZE(1..40)),	정류장ID
tpif	SubRouteIdentityNumber	UTF8String (SIZE(1..255)),	노선ID
tsfc	PTVehicleIDNumber	UTF8String (SIZE(1..255)),	차량ID
tpif	AccesspointArrivalTimeI	INTEGER (0..72000),	도착예정시간
tpif	LastBITIdentifyNumber	UTF8String (SIZE(1..255)) OPTIONAL,	(최근통과정류장) 진입시각
tsvh	LastBITZoneEntryTime	GeneralizedTime OPTIONAL,	(최근통과정류장) 진출시각
tsvh	LastBITZoneExitTime	GeneralizedTime OPTIONAL,	(최근통과정류장) 통과시간
tsvh	LastBITZoneTripTime	INTEGER (0..1200) OPTIONAL,	평균통행속도
tsvh	RollingAverageSpeedRate	INTEGER (0..300) OPTIONAL,	막차 정보
tsfc	LastPTVehicle	BOOLEAN OPTIONAL,	
tsfc	BUSRemainingSeatsNumber	INTEGER (0..512) OPTIONAL	버스잔여좌석정보

명칭	설명	단위
메시지 발생시각	해당시스템에 따라 정보가 생성된 시각	-
정류장ID	도착예정정보를 제공할 대중교통정류장의 고유식별자	-
노선ID	대중교통 노선의 고유식별자	-
차량ID	대중교통 사업자의 각 차량에 지정된 고유번호	-
도착예정시간	정보제공정류장에 도착하기까지 소요되는 통행시간	초
출발(최근통과)정류장ID	가장 최근에 통과한 정류장 식별자	-
(최근통과정류장)진입시각	가장 최근에 통과한 정류장영역으로 진입한 시각	시각
(최근통과정류장)진출시각	가장 최근에 통과한 정류장영역에서 진출한 시각	시각
(최근통과정류장)통과시간	가장 최근에 통과한 정류장영역에 머무른 시간, 정차시간	초
평균통행속도	대중교통차량의 최근 구간평균통행속도	km/h
막차정보	현재 진행하는 차량의 막차여부	-
버스잔여좌석정보	광역급행버스에서 사용되는 정보로써 버스잔여좌석정보	명

## 2) 정보생성

- 도착예정정보는 도착예정시간정보를 강제항목, 대중교통차량의 현재 위치정보를 권고항목으로 규약하고 있음
- 그리고 대부분의 사업사례에서 협역무선통신방식의 경우에는 이벤트정보에 기반한 대중교통차량의 링크통행시간과 노드에서의 정지, 정차시간 누적합계로 산출하고 있음
- 반면 광역무선통신 및 측위방식의 경우 이벤트정보로 부터는 현재 위치정보를, 정주기정보로 부터는 도착예정소요시간(통행소요시간) 정보를 산출하고 있음
- 그러므로 도착예정시간정보는 구현하는 시스템 방식 간에 입력되는 원본 데이터의 속성상 차이를 가지게 되나, 기술기준의 본문 규약에서는 도착예정소요시간(tpif—AccesspointArrivalTime)의 정수로만 한정하고 있으므로 시스템 방식으로 인한 정보항목의 구현상 문제는 발생되지 않음
- 그리고 대중교통차량의 현재 위치정보는 기존 사업사례에서 최근 통과정류장의 출발시각을 기준으로 위치정보를 산출하고 있으며(무정차의 경우도 동일), 정류장으로의 진입, 진출, 정차시간 등 모든 항목들이 권고항목으로 정의되어 있음

### 3) 상세설명

- 도착예정정보는 정보정의 시작 (ARRIVALPREDICTIONTIME DEFINITIONS AUTOMATIC TAGS ::= BEGIN) 으로부터 크게 도착예정시간정보와 대중교통 차량의 현재 위치정보로 구분되어 있음
- 도착예정정보는 실시간 수집정보에 의해 상시 산출되는 실시간 제공정보로서 동일한 위치에 존재하는 버스에 대하여서도 정보제공대상 정류장의 위치에 따라 다르게 표현되는 공간에 의존적인 정보임
- 따라서 정보제공 대상정류장을 기준으로 노선별 상행 및 하행의 방향을 고려한 특정버스의 도착예정시간 또는 현재 위치정보로 해석될 수 있음

<표 25> 도착예정정보 정의내용

ARRIVALPREDICTIONTIME DEFINITIONS AUTOMATIC TAGS ::= BEGIN	-
ArrivalPredictionTimeInfo ::= SEQUENCE	{
{	
tmsg–MessageGenerationTime	GeneralizedTime,
tpif–BITIdentifyNumber	UTF8String (SIZE(1..40)),
tpif–SubRoutelentityText	UTF8String (SIZE(1..255)),
tsfc–PTVehicleIDNumber	UTF8String (SIZE(1..255)),
tpif–AccesspointArrivalTime	INTEGER (0..72000),
tpif–LastBITIdentifyNumber	UTF8String (SIZE(1..255)) OPTIONAL,
tsvh–LastBITZoneEntryTime	GeneralizedTime OPTIONAL,
tsvh–LastBITZoneExitTime	GeneralizedTime OPTIONAL,
tsvh–LastBITZoneTripTime	INTEGER (0..1200) OPTIONAL,
tsvh–RollingAverageSpeedRate	INTEGER (0..300) OPTIONAL,
tsfc–LastPTVehicle	BOOLEAN OPTIONAL,
tsfc–BUSRemainingSeatsNumber	INTEGER (0..512) OPTIONAL
}	}
① : 도착예정정보정의 시작	
② : 메시지 생성시각	
③ : 정보제공정류장 고유ID	
④ : 운행계통ID	
⑤ : 대중교통차량ID	
⑥ : 도착예정소요시간	
⑦ : 최근통과정류장	
⑧ : 정류장 진입시각, 또는 통과시각	
⑨ : 정류장 진출시각	
⑩ : 정류장 정차시간	
⑪ : 구산평균통행속도	
⑫ : 막차여부	
⑬ : 버스잔여좌석정보	

### 5.5.3 운행계획정보(203)

#### 1) 개요

- 운행계획정보는 효율적인 버스운행관리를 위한 기초적인 정보로서, 최초 시스템 구축 시에는 기존 지방자치단체가 인허가한 운행계획을 근간으로 하나, 시스템 운영에 따라 사고, 고장 등의 돌발상황 발생 시나 이벤트 및 행사, 장기적인 공사 등을 지속적으로 업데이트해야 하는 정보임
- 따라서 운행계획정보는 자치단체 간 또는 상위 권역센터 간 정보연계에 있어서 필수적인 기초 정보가 되므로 기술기준으로 규정하고 있으며 크게 일반적인 식별정보와 계획정보로 구분할 수 있음

#### 203 운행계획정보

본 정보항목은 버스의 정직인 운행계획 정보를 규약함

정보명	운행계획정보
아키텍처상 정보명	운행계획정보
정보세부항목	노선기본정보(노선ID, 노선명칭, 기·종점정류장ID), 노선부가정보, 노선운행정보(첫차·막차 출발시각, 첨두/비첨두 배차간격), 차량운행횟수, 운행계획정보 갱신시각·갱신내용
설명	노선버스의 운행계획을 수립하거나 변경하는 때에는 등 정보를 미리 입력하여 센터~센터간 운행지시 및 조정과 협력, 이용자 여행스케줄에 활용

RouteSchedulePlanning ::= SEQUENCE {		정보의 이름/시각
tsmg—MessageGenerationTime	GeneralizedTime,	메세지발생시각
tpif—SubRouteIdentityText	UTF8String (SIZE(1..255)),	노선ID
tpif—SubRouteNameText	UTF8String (SIZE(1..255)) OPTIONAL,	노선명칭
tpif—RouteGuide	SEQUENCE OF RouteGuide,	노선부가정보
tpif—BeginBITIdentifyNumber	UTF8String (SIZE(1..40)) OPTIONAL,	기점 정류장ID
tpif—EndBITIdentifyNumber	UTF8String (SIZE(1..40)) OPTIONAL,	종점 정류장ID
scrn—TimeBeginTime	GeneralizedTime	첫차출발시각
scrn—TimeEndTime	GeneralizedTime	막차출발시각
tpif—PicktimeHeadwayTime	INTEGER (0..86400) OPTIONAL,	첨두시 배차간격
tpif—NonPicktimeHeadwayTime	INTEGER (0..86400) OPTIONAL,	비첨두시 배차간격
tsvh—PTVehicleFrequencyQuantity	INTEGER (0..1440) OPTIONAL,	차량운행횟수
tpif—RouteSchedulePlanningUpdate dTime	GeneralizedTime OPTIONAL,	운행계획정보 갱신시각
tpif—RouteSchedulePlanningUpdate dContents	UTF8String(SIZE(1..255)) OPTIONAL	운행계획정보 갱신내용
}		
RouteGuide ::=	UTF8String	

명칭	설명	단위
메시지 발생시각	해당시스템에 따라 정보가 생성된 시각	-
노선ID	버스정보시스템의 기반정보 구축 및 관리요령 중 노선기본정보 대중교통 노선의 고유식별자	-

명칭	설명	단위
노선명칭	버스정보시스템의 기반정보 구축 및 관리요령 중 노선기본정보 운행계통을 포함한 대중교통노선의 명칭	-
노선부가정보	버스정보시스템의 기반정보 구축 및 관리요령 중 노선부가정보 대중교통노선의 운행경로를 전부 나열함. 노선이 통과하는 모든 정류장의 목록을 설명	-
기점 정류장ID	버스정보시스템의 기반정보 구축 및 관리요령 중 노선기본정보 대중교통노선의 시작 정류장(기점)	-
종점 정류장ID	버스정보시스템의 기반정보 구축 및 관리요령 중 노선기본정보 대중교통노선의 마지막 정류장(종점)	-
첫차출발시각	버스정보시스템의 기반정보 구축 및 관리요령 중 노선운행정보 운행관리 및 정보제공을 위한 첫차의 기점출발 운행시각 정보	시각
막차출발시각	버스정보시스템의 기반정보 구축 및 관리요령 중 노선운행정보 운행관리 및 정보제공을 위한 막차 기점출발 운행시각 정보	시각
첨두시배차간격	버스정보시스템의 기반정보 구축 및 관리요령 중 노선운행정보 첨두시(최소) 노선의 배차간격	초
비첨두시배차간격	버스정보시스템의 기반정보 구축 및 관리요령 중 노선운행정보 비첨두시(최대) 노선의 배차간격	초
차량운행횟수	하루의 해당차량 운행횟수	회
운행계획정보갱신시각	운행계획정보가 최신 갱신된 시각	-
운행계획정보갱신내용	갱신시 바뀌는 정보항목 및 간략한 내용	-

## 2) 정보생성

- 운행계획정보는 실제 노선ID와 노선별 운행경로를 제외하고는 모든 정보항목이 권고항목으로 규정되어 있으나, 타 자치단체의 운행 계통을 파악하는 데 있어서 핵심적인 정보이므로 가급적 준수하는 것이 바람직 함
- 한편 운행계획정보의 특정상황에서 조정 등 업데이트는 운행지시정보를 통해서 가능하나, 근본적으로는 운수회사 또는 배차실에서 작성되는 실제 해당일의 배차계획표와 연동되도록 시스템을 구현하여 실제 배차계획정보와 운행계획정보가 최대한 일치하도록 운영되고 시민에게도 정보가 제공될 수 있어야 함
- 공동배차, 준공영제 등 버스에 배정되는 노선의 변동 유무를 막론하고 실제 배차계획정보와 운행계획정보를 일치하기 위해서는 버스가 출발하는 차고지, 회차지 등에서 노선번호를 부여하고 출차, 배차시간에 대한 검지 등 시스템적으로 자동화된 관리체계가 필요 함

- 이를 위해서는 광역무선통신은 물론 좁은 통신환경의 국부적인 통신체계를 도입 함으로써 출차하는 버스에 대한 노선번호 부여, 버스단말기 어플리케이션의 버 전관리, 노선, 정류장DB 등에 대한 원격업데이트가 가능하며, 도입되는 통신망의 속도에 따라서는 단말 운영체제까지도 업데이트가 가능하게 됨

### 3) 상세설명

- 운행계획정보는 정보정의 시작(TRANSITSCH DEFINITIONS AUTOMATIC TAGS ::= BEGIN)으로부터 크게 운행노선의 식별정보와 상세 계획정보로 구분되어 있음
  - 운행노선의 식별정보는 버스노선ID 및 설명, 명칭, 방향, 운행계통 ID로 정의되어있으며 노선별 상세 계획정보는 첫차, 막차시간과 첨두시 및 비첨두시 배차간격 등의 정보로 구성됨

### 〈표 26〉 운행계획정보 정의내용

TRANSITSCH DEFINITIONS AUTOMATIC TAGS ::= BEGIN	-
RouteSchedulePlanning ::= SEQUENCE	①
{	{
tsgm-MessageGenerationTime	GeneralizedTime,
tpif-SubRouteIdentityText	UTF8String (SIZE(1..255)),
tpif-SubRouteNameText	UTF8String (SIZE(1..255)) OPTIONAL,
tpif-RouteGuide	SEQUENCE OF RuoteGuide,
tpif-BeginBITIdentifyNumber	UTF8String (SIZE(1..40)) OPTIONAL,
tpif-EndBITIdentifyNumber	UTF8String (SIZE(1..40)) OPTIONAL,
scrn-TimeBeginTime	GeneralizedTime OPTIONAL,
scrn-TimeEndTime	GeneralizedTime OPTIONAL,
tpif-PicktimeHeadwayTime	INTEGER (0..86400) OPTIONAL,
tpif-NonPicktimeHeadwayTime	INTEGER (0..86400) OPTIONAL,
tsvh-PTVehicleFrequencyQuantity	INTEGER (0..1440) OPTIONAL,
tpif-RouteSchedulePlanningUpdatedTime	GeneralizedTime OPTIONAL,
tpif-RouteSchedulePlanningUpdatedContents	UTF8String (SIZE(1..255)) OPTIONAL
}	⑬ }
① : 운행계획정보 정의 시작	⑦ : 첫차 출발시각
② : 운행계획정보 메시지발생시각	⑧ : 막차 출발시각
③ : 노선 ID, 식별자	⑨ : 첨두시 배차간격
④ : 노선 명칭	⑩ : 비첨두시 배차간격
⑤ : 기점정류장 ID, 식별자	⑪ : 차량운행횟수, 빈도
⑥ : 종점정류장 ID, 식별자	⑫ : 운행계획정보가 최신된 갱신시각
	⑬ : 갱신시 바뀌는 정보항목 및 간략한 내용

## 5.5.4 운행지시정보(204)

### 1) 개요

- 운행지시정보는 운행계획에 따라 운행 중인 대중교통차량의 운행상태를 변경하기 위한 정보를 규약하고 있으며, 변경대상 운행상태로는 차간간격의 조정과 운행중단, 대차투입, 대기정차 등 서비스 지시코드로 구분하여 정의하고 있음

### 2) 정보생성

- 운행지시정보는 버스정보센터 운영자나 운수회사, 배차관리실에서 실시간으로 대중교통차량의 운전자에게 제공하는 운행상태 조정정보로서, 해당 대중교통차량의 차량, 도로 등 실시간 운행여건에 따라 변경상태가 작성될 수 있음

**204 운행지시정보**

본 정보항목은 특정 버스의 운행상태를 변경하기 위한 정보형식을 규약함

정보명	운행지시정보
아키텍처상 정보명	운행조정정보
정보세부항목	차량ID, 다음정차정류장ID, 차간거리조정, 운행지시정보
설명	긴급상황 정보에 따라 또는 운행계획을 변경함에 따라 보완·조정되는 버스운행 지시정보

```
OperationPlanning ::= SEQUENCE
{
    tsmg--MessageGenerationTimeGeneralizedTime,
    tsfc--PTVehicleIDNumberUTF8String(SIZE(1..255)),
    tpif--NextBitIdentifyNumberUTF8String(SIZE(1..40)),
    tpif--HeadwayTimeOffsetQuantityINTEGER(-600..600),
    scrn--ServiceDirectionCodeBIT STRING
    {
        break(0),
        commit--PTVehicle(1),
        stand-by(2)
    }OPTIONAL
}
```

**정보의 이름/시각**

**메시지발생시각**

**차량ID**

**다음정차정류장ID**

**차간거리조정**

**운행지시정보**

**운행중단**

**대차투입**

**대기정차**

**식별정보**  
**조정정보**

명칭	설명	단위
메시지발생시각	해당시스템에 따라 정보가 생성된 시각	-
차량ID	대중교통 사업자의 각 차량에 지정된 고유번호	-
다음정차정류장ID	해당노선의 다음정차 정류장 ID	-
차간거리조정	운행계획대비 차간간격의 조정	분
운행지시정보	긴급상황에 따른 운행지시사항 (코드)	-

### 3) 상세설명

- 정보정의 시작(TRANSITRUNCUT DEFINITIONS AUTOMATIC TAGS ::= BEGIN)으로부터 크게 운행지시대상 차량에 대한 식별정보와 운행조정정보로 구분되며, 운행조정정보 중 서비스지시코드는 권고항목으로 정의되어 있음
- 차간간격 조정수량(tpif-HeadwayTimeOffsetQuantity)은 분(min)단위 입력값으로 최대 10시간(600분)까지 증감이 가능하도록 정의되어 있음

<표 27> 운행지시정보 정의내용

TRANSITRUNCUT DEFINITIONS AUTOMATIC TAGS ::= BEGIN	-
OperationPlanning ::= SEQUENCE	{①}
{	{
tsmg—MessageGenerationTime	GeneralizedTime,
tsfc—PTVehicleIDNumber	UTF8String     (SIZE(1..255)),
tpif—NextBITIdentifyNumber	UTF8String     (SIZE(1..40)),
tpif—HeadwayTimeOffsetQuantity	INTEGER       (-600..600),
scrn—ServiceDirectionCode	BIT STRING
{	{
break	(0),
commit_PTVehicle	(1),
stand_by	(2)
} OPTIONAL	{②}
}	{③}
① : 운행지시정보 정의 시작	{④}
② : 운행지시정보 메시지발생시각	{⑤}
③ : 대중교통차량 ID, 식별자	{⑥}
④ : 차간간격조정수량(분)	{⑦}
⑤ : 서비스 지시코드,	{⑧}
{ ⑥ : 운행중단이면, “0”	}
⑦ : 대차투입이면, “1”	
⑧ : 대기정차이면, “2” }	

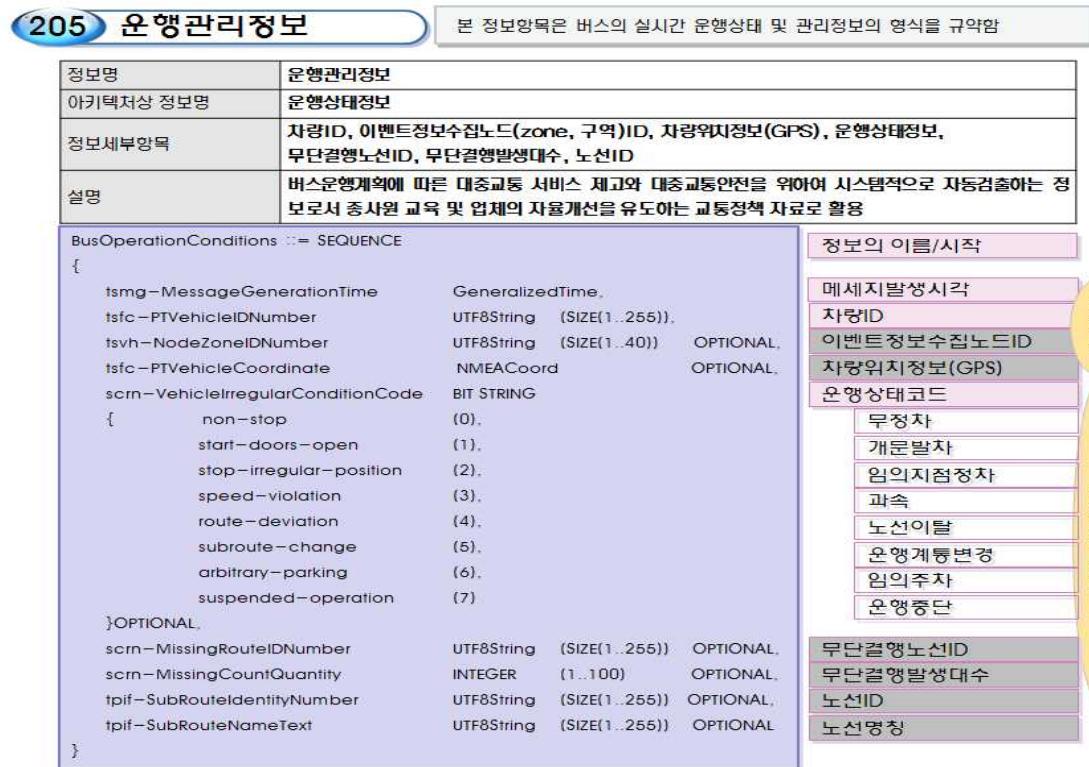
- 예를 들어, 2013년 12월 23일 오전 10:00 에 버스정보센터 운영자가 11-2번 노선의 1188번 버스에 차간거리를 현재보다 15분 단축시킬 것과 운행중단을 지시하였다면 메시지는 다음과 같이 구성됨

2013-12-23-10-00	1102-1188	6301005200	-15	0
메시지 생성시각	대중교통차량ID	도착예정정류장ID	차간간격조정량(분)	운행지시코드(운행중단)

## 5.5.5 운행관리정보(205)

### 1) 개요

- 운행관리정보는 대중교통차량이 운행 중 발생하는 실시간 운행상태를 규약하는 정보이며, 기술기준에서 제시하고 있는 운행관리 상태는 무정차, 개문발차, 임의지점정차, 과속, 노선이탈, 운행계통변경, 임의주차 등의 구분코드로 구성됨



명칭	설명	단위
메시지발생시각	해당시스템에 따라 정보가 생성된 시각	-
차량ID	대중교통 사업자의 각 차량에 지정된 고유번호	-
이벤트정보수집노드ID	대중교통 차량이 정규노선상을 운행할 때 경유하게 되는 노드영역(교차로, 정보수집지점, 정류장 등)의 고유번호	-
차량위치정보(GPS)	대중교통 차량의 GPS 위치정보. 비정상적 상황(위반 등)이 발생했을 때의 차량의 위치정보 수집	-
운행상태정보	대중교통차량의 비정상적 상황, 상태 지침 코드	-
무단결행노선ID	운행계획 변경없이 차량이 운행하지 않는 경우 해당 대중교통노선 고유식별자	-
무단결행발생대수	운행계획의 변경없이 차량이 운행하지 않는 횟수	대
노선ID	버스정보시스템의 기반정보 구축 및 관리요령 중 노선기본정보 대중교통 노선의 고유식별자	-
노선명칭	버스정보시스템의 기반정보 구축 및 관리요령 중 노선기본정보 운행계통을 포함한 대중교통노선의 명칭	-

- 운행관리정보는 단일 지자체사업에서 수집되어 위반정보 등에 대한 노선별 운행 관리 또는 운수회사에 대한 자동화된 모니터링 수단으로 활용되는 정보이며, 버스내단말기와 타코미터, 도어센서 등과 연동하여 측위정보와 함께 실시간으로 정보센터로 수집되는 가장 원시적인 정보임
- 센터간 대중교통차량의 위치정보는 NMEA 좌표정보값을 기준으로 교환하며, 그 내용은 ‘201. 버스위치정보’에서 규정한 내용을 따른

## 2) 정보생성

- 버스운행 수집정보인 이벤트정보(지점통과정보)와 정주기정보(실시간 위치정보)를 버스정보센터로 전송할 때 버스의 운행상태정보를 버스정보센터로 전송함으로써 운행관리(위반 단속 등) 목적으로 활용할 수 있음

버스수집정보형식 (차량->센터)	(1)	(2)	(3)	(4)
	데미타코드	버스위치정보	운행관리정보	긴급상황정보
버스제공정보형식 (센터->차량)	(1)	(2)	(3)	(4)
	데미타코드	도착예정정보	운행지시정보	긴급상황정보

(1) 수집주기, 이벤트, 정주기 등 정보특성을 구분하기 위한 데이터 코드  
 (2) 버스 통과위치 및 시각 등 실시간으로 버스위치 및 도착예정정보  
 (3) 대중교통차량의 도어센서, 운행기록계(타코미터) 등과 연동하여 생성되는 개문, 과속 등 운행상태와 관련한 정보(운행관리정보) 및 운행지시정보  
 (4) 차량고장, 사고, 돌발상황 등 운전자의 버스단말기 입력에 의존하여 버스정보센터로 송/수신되는 유형별 긴급상황정보

## 3) 상세설명

- 정보정의 시작 (TRANSCONDITION DEFINITIONS AUTOMATIC TAGS ::= BEGIN)으로부터 크게 대중차량 식별정보와 운행상태정보(위반정보항목 등)로 구분할 수 있음

#### 〈표 28〉 운행관리정보 정의내용

```

TRANSCONDITION DEFINITIONS AUTOMATIC TAGS ::= BEGIN
BusOperationConditions ::= SEQUENCE
{
    tsmg-MessageGenerationTime      GeneralizedTime,
    tsfc-PTVehicleIDNumber        UTF8String (SIZE(1..255)),
    tsvh-NodeZoneIDNumber         UTF8String (SIZE(1..40)) OPTIONAL,
    tsfc-PTVehicleCoordinate      NMEACoord           OPTIONAL,
    scrn-VehicleIrregularConditionCode BIT STRING
{
    non-stop                      (0),
    start-doors-open              (1),
    stop-irregular-position       (2),
    speed-violation              (3),
    route-deviation               (4),
    subroute-change                (5),
    arbitrary-parking             (6),
    suspended-operation          (7),
}OPTIONAL,
    scrn-MissingRouteIDNumber     UTF8String (SIZE(1..255)) OPTIONAL,
    scrn-MissingCountQuantity     INTEGER   (SIZE(1..100))  OPTIONAL,
    tpif-SubRouteIdentityNumber   UTF8String (SIZE(1..255)) OPTIONAL,
    tpif-SubRouteNameText         UTF8String (SIZE(1..255)) OPTIONAL
}

```

- ① : 운행관리정보 정의 시작
  - ② : 대중교통차량 ID, 식별자
  - ③ : 통과교차로/정류장 ID, 식별자
  - ④ : 차량비정상상태코드,
  - ⑤ : GPS 위치정보
  - {     ⑥ : 무정차이면 “0”,                      ⑦ : 개문발차이면 “1”,                      ⑧ : 임의지점정차이면 “2”  
        ⑨ : 과속/속도위반이면 “3”,   ⑩ : 노선이탈이면 “4”,                      ⑪ : 운행계통변경이면 “5”  
        ⑫ : 임의주차이면 “6”,                      ⑬ : 운행중단이면 “7” }
  - ⑭ : 결행발생노선 ID, 식별자(권고항목)
  - ⑮ : 노선별 결행발생누계횟수(권고항목)
  - ⑯ : 노선의 고유식별자
  - ⑰ : 운행계통을 포함한 대중교통노선의 명칭

- 대중교통차량의 비정상 상태코드(scrn-VehicleIrregularConditionCode)는 0~6의 정수로 정의되며, 시스템에서 상태 판단을 위한 구체적 예시는 다음 표와 같음

<표 29> 운행관리정보의 상태코드별 판단방법론

코드	항목	판단방법	판단근거(예시)
0	무정차	특정 버스의 정류장 출발이벤트가 수집되었으나, 버스가 개문되지 않고 속도가 일정속도 이하로 떨어지지 않을 경우	정류장 통과시 속도 > 4km/시, 문열림 = FALSE
1	개문발차	특정버스의 정류장 도착이벤트가 수집되었으나, 버스가 폐문되지 않는 상태에서 출발이벤트가 수집되는 경우	정류장 출발시 문열림 = TRUE
2	임의지점 정차	여객자동차운수사업법 제28조 1항에 근거, 정류장이 아닌 지점에서 여객을 승·하차 시키는 경우	이벤트정보수집 없이 출입문이 개·폐
3	과속/ 속도위반	운행기록계(타코미터) 또는 GPS에 의한 일정시간 평균산출속도가 도로의 제한속도를 초과하는 경우	운행속도 ≫ 제한속도
4	노선이탈	버스단말기의 입력노선DB 값과 GPS좌표에 의한 위치좌표가 일정시간 이상 현저하게 차이가 나는 경우	수집주기5회 이상,  오차거리  ≫ 100m
5	운행계통 변경	여객자동차운수사업법시행령 제2조 2항에 근거, 기점, 종점 및 기종점간 운행거리, 경유지, 횟수, 대수, 운행간격 등에 일부 변경이 발생하는 경우	측위좌표를 이용한 관제 (노선이탈 참조)
6	임의주차	버스운행 중에 단말기 전원이 On, Off 되거나, 자체, 정체가 아닌 상황에서 도로 특정지점에 일정시간 이상 머무르는 경우	전원상태변동=TRUE  정차시간  ≫ 10분
7	운행중단	정상적으로 단말기 전원이 Off 되거나, 노선의 종점지점에서 머무르는 경우	노선의 종점에 위치한 경우

- 한편 결행발생의 판단은 운행계획 대비 실제 해당일의 배차내역의 비교를 통해서 판단이 가능한 항목으로서, 센터운영자는 노선별 발생횟수를 누계함으로써 월별, 분기별 통계적인 분석이 가능하게 됨

## 5.5.6 긴급상황정보(206)

### 1) 개요

- 긴급상황정보는 운행 중인 대중교통차량에게 발생되는 돌발상황에 대한 정보를 규약하기 위한 항목으로서 특정 대중교통차량의 현재 위치와 돌발상황의 발생위치, 돌발상황 유형 및 진행단계에 대한 정보로 구성되어 있음

#### 206 긴급상황정보

본 정보항목은 버스의 실시간 긴급, 돌발상황 정보를 규약함

정보명	긴급상황정보		
아키텍처상 정보명	돌발상황정보완정보		
정보세부항목	차량ID, 노선ID, 이벤트정보수집노드(zone, 구역)ID, 차량위치정보(GPS), 발생위치(도로명칭, 관련교차로, 돌발상황발생위치설명), 돌발상황발생시각, 돌발상황부연설명, 돌발상황유형, 돌발상황유형부연설명, 돌발상황긴급정도, 돌발상황긴급정도 부연설명		
설명	운행중인 대중교통에 발생하는 돌발상황 정보		

CurrentIncidentsStatus ::= SEQUENCE {	정보의 이름/시작
tsmg—MessageGenerationTime	GeneralizedTime.
tsfc—PTVehicleIDNumber	rUTF8String (SIZE(1..255)) OPTIONAL,
tpif—SubRouteIdentityNumber	UTF8String (SIZE(1..255)) OPTIONAL,
tsvh—NodeZoneIDNumber	UTF8String (SIZE(1..40)) OPTIONAL,
tsfc—PTVehicleCoordinate	NMEACoord OPTIONAL,
evnt—LocationRoadwayNameText	UTF8String (SIZE(1..64)) OPTIONAL,
evnt—LocationCrossStreetBeginText	UTF8String (SIZE(1..64)) OPTIONAL,
tfmg—IncidentLocationText	UTF8String (SIZE(1..255)) OPTIONAL,
tfmg—IncidentGeneratedTime	GeneralizedTime,
tsin—IncidentDescriptionLongText	UTF8String (SIZE(1..255)) OPTIONAL,
tfmg—IncidentType	BIT STRING
{	
ptvehicleBreakDown	(0),
ptvehicleAccident	(1),
emergencySituation	(2),
incidentInPtvehicle	(3),
passingControl	(4),
notPermitOperation	(5) }OPTIONAL,
tsin—IncidentTypeOtherText	UTF8String (SIZE(1..255)) OPTIONAL,
tsin—IncidentLevelCode	ENUMERATED
{	
incidentDetected	(0),
confirmedAndResponding	(1),
clearedAndRecovering	(2),
overAndDone	(3) }OPTIONAL,
tsin—IncidentLevelOtherText	UTF8String (SIZE(1..255)) OPTIONAL
}	

차량 정보

발생 위치 정보

긴급 상황 정보

명칭	설명	단위
메시지 발생시각	해당시스템에 따라 정보가 생성된 시각	-
차량ID	돌발상황을 관측하거나, 돌발상황에 있는 차량ID	-
노선ID	대중교통 노선의 고유식별자	-
이벤트정보수집노드ID	대중교통 차량이 정규노선상을 운행할 때 경유하게 되는 노드영역(교차로, 정보수집지점, 정류장 등)의 고유번호	-
차량위치정보(GPS)	대중교통 차량의 GPS 위치정보	-
도로명칭	돌발상황이 발생한 도로의 명칭	-
관련교차로	돌발상황이 발생한 시점과 관련된 교차로 명칭	-
돌발상황발생위치	돌발상황발생의 위치	-
돌발상황발생시각	돌발상황발생의 시각	-
돌발상황유형	발생된 돌발상황 유형 분류 코드	-
돌발상황유형부연설명	돌발상황에 대한 문자 설명의 긴 표현.	-
돌발상황긴급정도	긴급상황에 대한 정도로서 추가 대처방안을 알려주기 위한 정보	-
돌발상황긴급정도 부연설명	긴급상황에 대한 정도의 부연설명	-

- 또한 긴급상황정보는 도로상에 발생하는 사고, 고장, 긴급상황, 통행규제 등 기본교통정보교환 기술기준에서 규정하고 있는 돌발상황정보의 형식을 기반으로 하고 있으며 센터간 정보교환 측면에서 활용가치가 비교적 높은 정보항목임
- 센터간 대중교통차량의 위치정보는 NMEA 좌표정보값을 기준으로 교환하며, 그 내용은 ‘201. 버스위치정보’에서 규정한 내용을 따른

## 2) 정보생성

- 긴급상황정보는 버스정보센터에서 버스차내단말기로 전송하여 인접도로에 발생되는 긴급상황을 알리기 위한 목적으로 활용될 수 있으나, 버스운전자에 의한 입력, 수집정보로 구현되는 실정임
- 따라서 버스정보(메시지발생시각, 대중교통차량 ID, 운행계통 ID) 및 위치정보(통과노드 고유 ID, 위도, 경도, 도로명)는 정보전송시점을 기준으로 자체적으로 생성되는 정보이며, 긴급상황 유형, 진행과정은 버스운전자의 입력에 의해서 수집되는 정보임

### 3) 상세설명

<표 30> 긴급상황정보 정의내용

TRANSITINCIDENT DEFINITIONS AUTOMATIC TAGS ::= BEGIN	-
CurrentIncidentsStatus ::= SEQUENCE	① {
{	
tsmg–MessageGenerationTime GeneralizedTime,	②
tsfc–PTVehicleIDNumber                          UTF8String            OPTIONAL,	③
tpif–SubRouteIdentityText                        UTF8String            OPTIONAL,	④
tsvh–NodeZoneIDNumber                          UTF8String            OPTIONAL,	⑤
tsfc–PTVehicleCoordinate                        NMEACoord            OPTIONAL,	⑥
evnt–LocationRoadwayNameText                  UTF8String            OPTIONAL,	⑦
evnt–LocationCrossStreetBeginText             UTF8String            OPTIONAL,	⑧
tfmg–IncidentLocationText                      UTF8String            OPTIONAL,	⑨
tfmg–IncidentGeneratedTime                     GeneralizedTime,	⑩
tsin–IncidentDescriptionLongText             UTF8String            OPTIONAL,	⑪ {
tfmg–IncidentType                                BIT STRING {	⑫
ptvehicle_break_down                        (0),	⑬
ptvehicle_accident                          (1),	⑭
emergency_situation                        (2),	⑮
incident_in_ptvehicle                      (3),	⑯
passing_control                             (4),	⑰
not_permit_operation                        (5) }                OPTIONAL,	⑱ {
tsin–IncidentTypeOtherText                     UTF8String            OPTIONAL,	⑲ {
tsin–IncidentLevelCode                         ENUMERATED {	⑳
incidentDetected                            (0),	㉑
confirmedAndResponding(1),	㉒
clearedAndRecovering(2),	㉓
overAndDone                                (3) }                OPTIONAL,	㉔
tsin–IncidentLevelOtherText                   UTF8String            OPTIONAL	}
}	}
① : 긴급상황정보 정의 시작,	① : 긴급상황정보 메시지생성시각
② : 대중교통차량 ID, 식별자	③ : 노선 ID, 식별자(해당 노선ID)
④ : 통과노드고유 ID, 식별자(최근 통과교차로 또는 정류장 ID)	
⑤ : ②에서 지시하는 대중교통차량의 GPS위치	
⑥,⑦ : 긴급상황이 발생한 도로명칭과 관련된 교차로명칭	
⑧,⑨ : 긴급상황의 발생위치명칭 및 발생시각	
⑩ : 긴급상황에 대한 설명(주석)	
⑪ : 긴급상황 발생유형	
{   ⑫ : 자차고장,   ⑬ : 자차사고,   ⑭ : 긴급상황,	
⑮ : 차내사고,   ⑯ : 통행규제,   ⑰ : 운행불가   }	
⑯ : 긴급상황 발생유형에 대한 설명(주석)	
⑰ : 긴급상황 발생 수준(단계구분코드)	
{   ⑱ : 사고감지단계,   ⑲ : 확인 및 대응조치단계,	
⑳ : 처리 및 복구단계,   ㉑ : 상황종료단계   }	
㉒ : 긴급상황 발생수준에 대한 부연설명(주석)	

- 정보정의 시작(TRANSITINCIDENT DEFINITIONS AUTOMATIC TAGS ::=BEGIN)으로부터 차량, 상황발생 위치 등의 식별정보와 상황유형, 진행과정 등 긴급상황 발생정보로 구분됨
- 식별정보는 차량, 노선, 통과노드 ID 및 차량의 위치 등 차량정보와 도로 및 교차로, 위치, 시각 등 긴급상황이 발생한 공간적 위치와 시각정보로 정의되어 있음
- 한편 긴급상황정보는 긴급상황 유형을 자체 차량고장 및 사고발생, 긴급상황, 차내사고, 통행규제(버스우선신호 처리 등), 운행불가로 구분하고 있으며, 긴급상황 수준 구분은 사고감지단계, 확인 및 대응단계, 처리 및 복구단계, 상황종료단계로 구분하고 있음
- 긴급상황 유형구분 중 긴급상황(emergency\_situation)은 고장, 사고, 차내사고, 통행규제, 운행불가 등 이외의 상황에 대해서도 포괄적으로 적용 가능한 옵션임

## 6. DSRC를 이용한 ETCS의 정보교환 기술기준

### 6.1 개요

- 전자지불시스템 구축 활성화를 위해 자동요금징수에 필요한 표준으로 「근거리 전용통신(DSRC)를 이용한 자동요금징수시스템(ETCS)의 정보교환 기술기준(노변-단말간)」을 국토해양부(現 국토교통부)에서 제정·고시함
  - 제정 2006년 7월 31일, 국토해양부 고시 제2006-304호
  - 최종개정 2013년 5월 16일, 국토교통부 고시 제2013-251호 (일부개정)

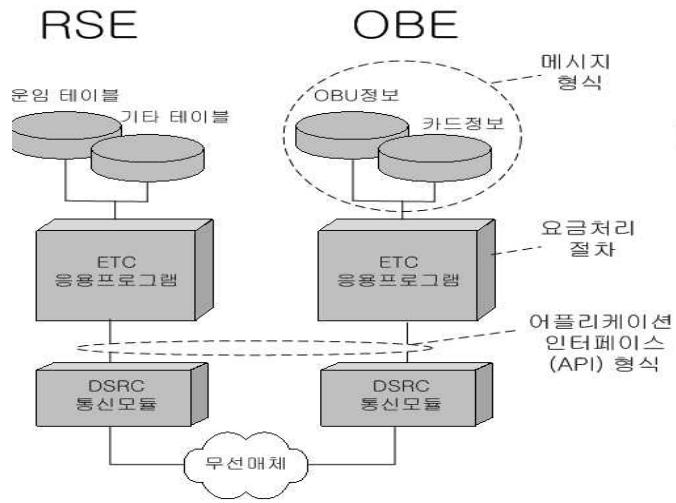
### 6.2 기술기준의 목적 및 범위

#### 6.2.1 기술기준의 목적

- 교통시설의 이용요금을 현금·수표 등이 아닌 전자거래로 지불하는 징수체계(이하 ‘ETCS’라 함, Electronic Toll Collection System)의 핵심기술인 근거리 전용 무선통신(이하 ‘DSRC’라 함, Dedicated Short – Range Communication) 규약의 표준화된 응용 인터페이스 방식 및 정보형식을 정하여 요금지불로 인한 정산지체를 최소화하기 위한 것임

#### 6.2.2 기술기준의 범위

- DSRC의 기반인 노변장치(이하 ‘RSE’라 함, Road Side Equipment)와 차량 단말기(이하 ‘OBE’라 함, On Board Equipment) 사이에서 요금을 주고받는 시스템에 적용됨
- DSRC의 매체방식 즉 통신 물리계층인 적외선, RF 등의 차이에 관계없이 적용될 수 있음
- 고속도로 등 유료도로의 요금소 뿐만 아니라 주차장, 주유소 등 차량을 이용하는 교통시설 또는 서비스에 대한 요금징수에 사용 가능함
- 기술기준의 세부 범위는 아래 그림의 3가지 부분에 대하여 정의할 수 있음



<그림 24> 기술기준의 범위

- DSRC 응용 인터페이스(API) 형식
- ETC 응용프로그램의 요금처리 절차 (ETC–Debit Transaction flow)
- ETC 응용프로그램이 이용하는 정보 형식 (Message format)

## 6.3 기술기준의 구성

### 6.3.1 명령어

- DSRC 응용 인터페이스(API)는 ISO 14906(KS X ISO 14906) 표준에서 정의하는 5가지 기본적인 명령어를 사용함

명령어 종류	API 명령어	설명
초기화 명령	Initialize	OBE(OBU와 카드)를 초기화
정보 읽기 명령	Get	OBE의 정보를 읽기
정보 쓰기 명령	Set	OBE에 정보를 쓰기
동작 수행 명령	Action	OBE에 특별한 동작 수행을 요청
이벤트 정보전달	Event Report	OBE의 상태를 문의하거나 지정

- 동작 수행 명령인 Action은 ETCS에 적합하도록 15개의 서브함수로 정의하고 있으며 본 기준에서는 아래 3가지 서브함수를 사용
  - Action–Debit : OBE에 요금차감을 수행하도록 요청
  - Action–Set–Secure : OBE에 정보를 안전하게 쓰기

### 6.3.2 요금처리 절차(ETC-Debit Transaction flow)

- ETC 응용프로그램의 요금처리 절차(ETC-Debit Transaction flow)는 아래와 같이 3단계로 구분하여 수행됨
  - 초기화 절차(Initialization phase) : OBE가 카드를 초기화하고 RSE에게 통신영역의 진입을 통보하는 절차
  - 징수 절차(Transaction phase) : RSE가 카드 및 OBU내에 기록된 기본발행정보 및 교통 관련 추가정보를 읽고 요금을 계산한 후, 카드의 요금차감을 수행하는 절차
  - 후속처리 절차(Post transaction phase) : 요금차감 후 처리결과를 OBU 및 카드에 기록하는 절차

### 6.3.3 정보형식(Message format)

- ETC 응용프로그램이 이용하는 정보형식(Message format)은 OBU와 카드에 대한 발행정보 및 어플리케이션 정보의 포맷을 의미하는 것으로, 정보는 속성파일(Attribute) 형태로 지정됨

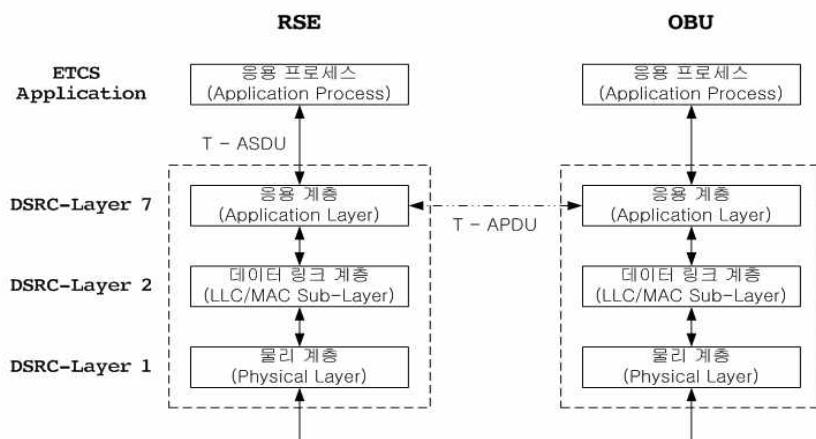
### 6.3.4 다양한 카드의 처리 방안

- 고속으로 주행 중인 차량에 대하여 실시간 요금징수가 가능한 고속처리 알고리즘(FTA, Fast Transaction Algorithm)을 탑재하고 있고 다양한 요금체계(개방식, 폐쇄식, 접속식)를 반영하도록 설계된 카드 및 OBU에 대하여 최적화되도록 정의함
- 즉, 기존의 교통카드(예: 버스카드 등)나 금융카드(예: 후불신용카드)등 다른 카드를 사용하는 경우라면 RSE가 표준에 기술하는 절차에 따라서 요금징수를 실행할 수 없기 때문에, RSE는 Action 명령어 중 Transfer-channel 서브함수를 이용하여 카드요금을 차감하도록 함

- Transfer-channel 서브함수를 이용해 카드요금을 차감하게 되면 OBU는 단순히 무선통신만을 수행하게 되어 실질적 요금처리는 기존 카드의 징수처리절차에 따라 RSE가 카드를 직접 제어하도록 함
- 이 경우 카드처리 절차의 복잡도 및 수행능력으로 인해 발생되는 요금징수 처리 시간의 지연에 대해서는 적절한 보완처리를 수행하여야 함
- 향후 ETCS 제휴카드는 최적화된 ETCS 지불 프로토콜을 탑재하여 RSE가 동일한 표준에 따른 요금징수를 실행할 수 있도록 하여야 함
- 교통카드와 금융카드 등 모든 카드를 사용하기 위해 OBU에서는 사용자가 ETCS 제휴카드 이외의 카드를 사용할 경우 새로운 EID를 생성하며, 카드의 구분을 위한 Context Mark를 파라미터로 할당함
- Context Mark는 KS X ISO 14906:2005의 8.2 Contract 데이터 그룹을 참조하여 국토교통부에서 부여하고 관리도록 함
- 단순화되거나 다른 속성(Attribute)을 이용하여 요금징수를 처리하고자 하는자는 희망사항을 사업계획에 추가하여 국토교통부의 승인을 득하여야 하며, 국토교통부는 해당사항을 반영하여 적절한 절차에 따라 기술기준을 개선할 수 있음

## 6.4 응용인터페이스

### 6.4.1 DSRC와 ETCS 응용프로세스의 관계



<그림 25> DSRC와 ETCS 응용프로세스의 관계

- 본 기술기준에서 정의하는 ETCS 응용인터페이스는 DSRC 응용계층의 서비스 프리미티브를 이용하여 요금의 자동인식 및 징수에 대한 정보를 수집, 처리, 제공하는 서비스를 의미함
- 그림은 DSRC의 응용계층과 ETCS 응용프로세스간 서비스 프리미티브를 이용한 정보교환을 의미함

#### 6.4.2 서비스 프리미티브

##### 1) 서비스 프리미티브를 이용한 정보교환 기능

- INITIALIZATION : 사용자가 INITIALIZATION 서비스를 호출하면, 통신이 이루어지지 않은 OBU와 RSE간에 통신 초기화를 시도하게 됨
- GET : 사용자가 GET 서비스를 호출하면 상대방의 정보검색을 통해 필요한 정보를 얻을 수 있으며, 이 서비스는 확인 모드에서만 서비스 요청이 가능하고 요청에 대해서는 반드시 응답하여야 함
- SET : 사용자가 SET 서비스를 호출하면 상대방의 정보를 수정함
  - 확인/비확인 모드 둘 다 서비스 요청이 가능하며, 확인 모드에서는 요청에 대해 반드시 응답
  - 브로드캐스트와 같은 특별한 상황에서는 비확인 모드에서 서비스 요청만 수행 가능
- ACTION : 사용자가 ACTION 서비스를 호출하면 상대방으로 하여금 원하는 동작을 수행하도록 할 수 있음
  - 확인/비확인 모드 둘 다 서비스 요청이 가능하며, 확인 모드에서는 요청에 대해 반드시 응답
  - 브로드캐스트와 같은 특별한 상황에서는 비확인 모드에서 서비스 요청만 수행 가능
- EVENT-REPORT : 사용자가 EVENT-REPORT 서비스를 호출하면 상대방에 이벤트를 보내고, 이 서비스는 확인/비확인 모드 둘 다 서비스 요청이 가능하며 확인 모드에서는 요청에 대해 응답해야 함

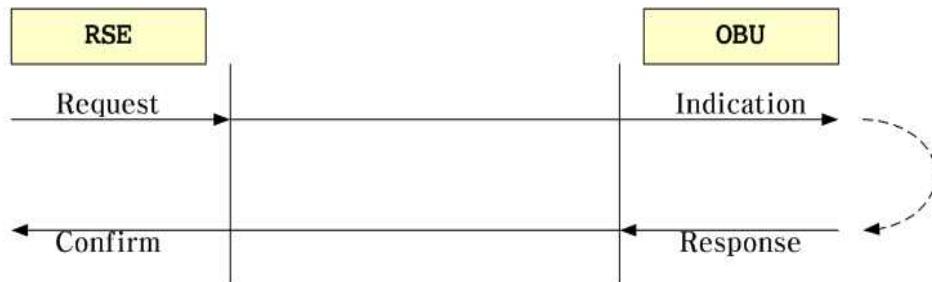
## 2) ETCS에서의 서비스 프리미티브 사용

- INITIALIZATION 서비스는 다음에 대하여 ETC 초기화 매커니즘을 위해 사용됨
  - (De-)Register Application RSE (RSE 측) : 제공 가능한 서비스를 RSE 응용계층에 등록하고 삭제하는 프리미티브
  - Notify Application Beacon (RSE 측) : 기지국의 초기화 커널 요소가 VST를 수신한 후 DSRCApplianceEntityID에 대해 등록된 요소를 Application에게 통보하는 프리미티브
  - Ready Application (RSE 측) : RSE에 Ready Application을 송신하면 RSE에서 초기화 커널 요소의 ApplicationList 엔트리가 비어있는 경우 상대 초기화 커널 요소로 Release를 전송하여 서비스 종료명령을 내리게 하는 프리미티브
  - (De-)Register Application OBE (OBE 측) : OBE에서 서비스를 종료할 때 해당 서비스를 OBE 응용 계층 목록에 등록하고 삭제하는 프리미티브
  - Notify Application Vehicle (OBE 측) : OBE의 응용계층이 BST를 수신한 후 OBE Application에게 등록된 요소를 통보하는 프리미티브
- GET 서비스는 ETC의 속성을 검색하는데 이용함
- SET 서비스는 ETC의 속성을 수정하는데 이용함
- ACTION 서비스는 TRANSFER\_CHANNEL, SET\_MM, ECHO, DEBIT등과 같은 ETCS 응용처리를 지원하기 위해 필요한 부가된 ETCS 특별기능을 구현하는데 이용함

## 3) ETCS 서비스 프리미티브를 위한 DSRC 응용계층

- ETCS는 DSRC 응용계층과의 연결을 통해 교환되는 서비스 프리미티브를 이용하여 기능을 구현하며 이는 다음과 같은 기본 패턴을 따름
  - 요구(xxx.Request) : 서비스 이용자가 기능을 호출하는 프리미티브로 본 기준에서는 RSE 응용의 DSRC 응용계층에 의해 수행
  - 지시(xxx.Indication) : 서비스 공급자가 기능을 호출하기 위해 또는 서비스 접근점에서 기능이 호출되었음을 지시하기 위해서 사용되는 프리미티브로 본 기준에서는 DSRC 응용계층의 OBU 응용에 의해 발행

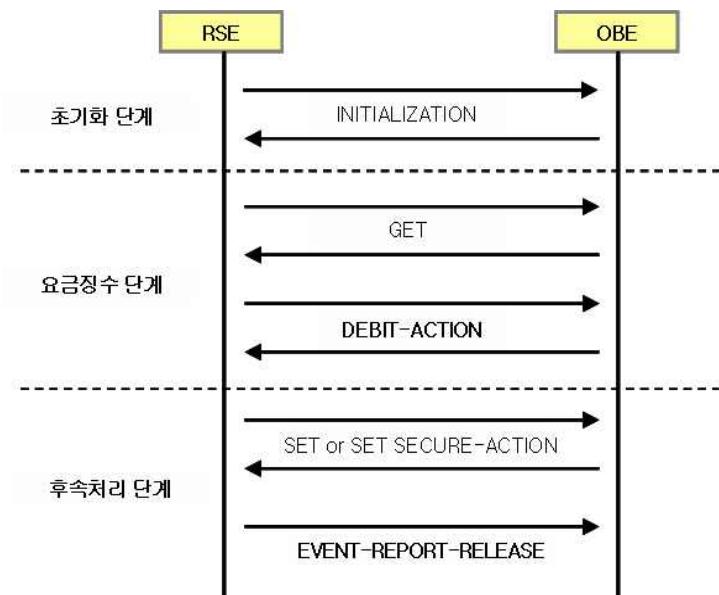
- 응답(xxx.Response) : 서비스 이용자가 지시에 의하여 이미 호출된 기능을 완료 시키기 위해서 사용되는 프리미티브로 OBU 응용의 DSRC 응용계층에 의해 수행
- 확인(xxx.Confirm) : 서비스 공급자가 요구에 의하여 이미 호출된 기능을 완료 시키기 위해서 사용되는 프리미티브로 DSRC 응용계층의 RSE 응용에 의해 발행



<그림 26> 초기서비스 교환의 논리적 절차

## 6.5 데이터를 이용한 요금징수절차

- ETCS를 이용한 요금징수절차는 아래 그림과 같이 초기화, 요금징수, 후속처리의 3단계로 구성됨



<그림 27> ETCS 요금징수절차 기본 구성

- 초기화 단계 : RSE가 OBE에 대해서 ID, 카드와 트랜잭션 등에 대한 정보를 요청하고 이에 대한 응답을 체크함으로써 요금징수가능 여부를 판단하는 역할을 수행

- 요금징수 단계 : RSE가 OBE에 대해서 카드와 카드소지자, 최근 트랜잭션 정보 등을 요청하고 이에 대한 응답을 확인 한 후에 해당 OBE를 통하여 카드의 금액을 차감하고 처리여부에 대한 응답을 확인하는 역할을 수행함
- 이 요금징수과정에서는 요금관련 데이터 이외에 적절한 보안 데이터 포함
- Debit Action Request Parameter

```
DebitRq ::= SEQUENCE
{
    fee                  PaymentFee,
    nonce               OCTET STRING,
    keyRef              INTEGER           (0..255)
}
```

```
PaymentFee ::= SEQUENCE
{
    paymentFeeAmount    INTEGER(0..65535),
    paymentFeeUnit      INTEGER(0..65535)
}
```

필드	길이	설명
차감 요금	4 바이트	2바이트 = 차감 요금 2바이트 = 단위 환산 <ul style="list-style-type: none"> <li>. 0xxx(hex) = 단위 환산 없음</li> <li>. 1xxx(hex) = 10:1</li> <li>. 2xxx(hex) = 100:1</li> <li>. 3xxx(hex) = 1000:1</li> <li>. 4xxx(hex) = / 10</li> <li>. 5xxx(hex) = / 100</li> <li>. 6xxx(hex) = / 1000</li> <li>. 7xxx(hex) = / 10000</li> <li>. 8xxx(hex) = / 100000</li> </ul> xxx 는 ISO 4217에서 정의하는 화폐단위(BCD)
파라미터 길이	1 바이트	값 = 24
파라미터 내용 (nonce)	24 바이트	3 바이트 = PSAM 공급자 ID 8 바이트 = PSAM ID 4 바이트 = PSAM 거래일련 번호 4 바이트 = S2 5 바이트 = 예비
파라미터 내용(keyref)	1 바이트	기본값 = 0

- Debit Action Response Parameter

```
DebitRs ::= SEQUENCE
{
    debitResult          ResultFin          ,
    debitAuthenticator  OCTET STRING
}
```

```
ResultFin ::= OCTET STRING(SIZE(1))
```

필드	길이	설명
차감 결과	1 바이트	0x00 OK 0x10 규정되지 않은 에러 0x20 카드처리 타임아웃 0x02 잔액부족 0x05 S2서명 인증에러
파라미터 길이	1 바이트	값 = 4
차감 인증자	4 바이트	S3

- EVENT-REPORT-RELEASE Request Parameter

```
Event-Report-Request ::= SEQUENCE
{
    mode           BOOLEAN,
    eid            Dsrc-EID,
    eventType      EventType,
    accessCredentials[] OCTET STRING(SIZE(0..127,...)) OPTIONAL,
    eventParameter Container OPTIONAL,
    iid            Dsrc-EID[] OPTIONAL
}
```

```
EventType ::= INTEGER
{
    release[](0)           EventType
    -- (1..118)   ISO/CEN 이용을 위해 예약
    -- (119..127)  민간 이용을 위해 예약
}(0..127,...)
```

※ I-Kernel을 통해서 처리할 경우

필드	길이	설명
이벤트 종류	1 바이트	값 : 00 (Release)
파라미터	0 바이트	Empty

※ I-Kernel을 통하지 않고 응용인터페이스 또는 T-Kernel을 통해 처리할 경우

필드	길이	설명
파라미터 갯수	1 바이트	1
파라미터 타입	1 바이트	2
파라미터 길이	1 바이트	1
파라미터 내용	1 바이트	Transaction OK = 0x00 Transaction NG 1) 카드없음 = 0x01 2) 잔액부족 = 0x02

- 후속처리 단계 : RSE가 OBE에 대해서 최근 트랜잭션 정보와 함께 거래내역과 거래종료에 대한 정보를 요청 및 확인함으로써 최종 요금징수여부를 체크하고 처리결과를 해당 OBE에 전송하는 역할을 수행

### 6.5.1 ETCS 요금징수절차의 세부 기능

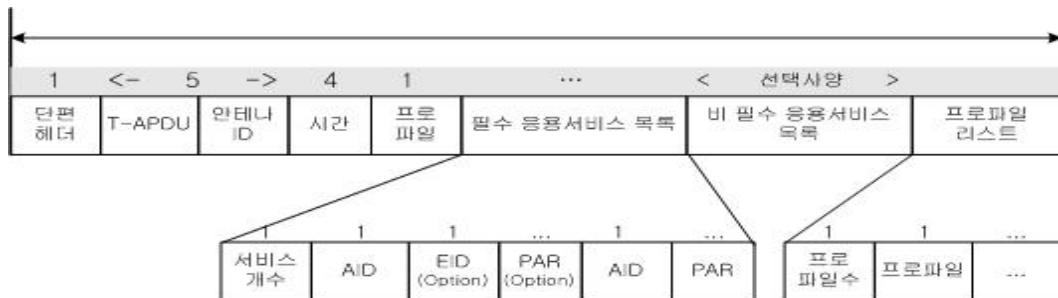
- 본 기술기준은 ASN.1(KS X ISO 8824-1:2004)을 이용한 PER(KS X ISO 8825-2:2004)의 규칙을 적용하며, DSRC를 이용한 ETCS 기능(Function)의 구조는 아래와 같음



※ APDU : 응용 서비스 프로토콜 데이터 단위(Application Protocol Data Unit)

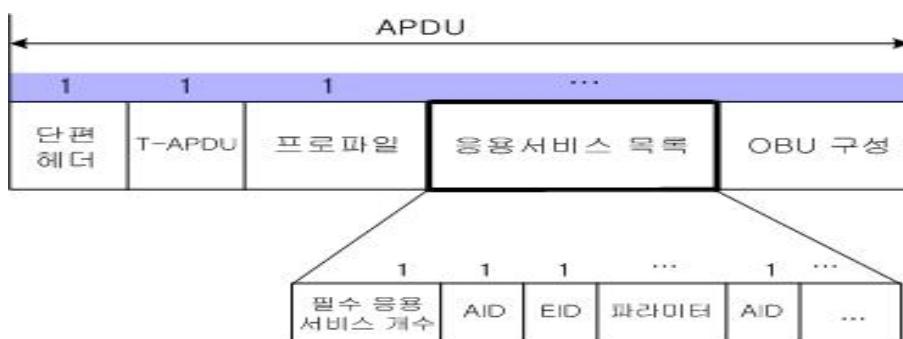
범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	APDU의 단편정보 <sup>주1)</sup> 를 나타내며, 단편헤더의 x3~x6(PDU number)는 프레임 번호를 나타냄
T-APDU	4X (hex)	응용 서비스의 종류 의미 SET.Request
	5X (hex)	SET.Response
	6X (hex)	GET.Request
	7X (hex)	GET.Response
	0X (hex)	ACTION.Request
	1X (hex)	ACTION.Response
	2X (hex)	EVENT-REPORT.Request
	3X (hex)	EVENT-REPORT.Response
	8X (hex)	INITIALIZATION.Request
	9X (hex)	INITIALIZATION.Response
EID	XX(hex)	서비스 대상 방법(수단)ID(EID, Element ID) <sup>주2)</sup> 으로 초기화(INITIALIZATION) 응용 서비스를 제외한 모든 응용 서비스에 적용되는 값
실제 응용 서비스 관련 데이터 내용:		
속성 ID 목록		속성 ID 목록
속성 목록		속성 목록
파라미터 목록		파라미터 목록
...		별도 데이터 구조
리턴 값	XX	리턴 값
주1) 단편정보는 통신에서 일반적으로 얘기하는 패킷 또는 프레임의 순서를 나타내는 정보로써 x3에서부터 차례로 증가시켜 번호를 부여한다.		
주2) EID는 서비스 장비의 이용 가능한 서비스 방법(수단)에 대한 ID를 나타내는 것으로 "0" 이외의 값을 할당하여 사용한다.(EID=0, 시스템으로 예약) EID는 OBE내에서 Session이 설정되었을 때 고유 번호를 OBE가 자체적으로 부여하며, 계약관련사항은 Context Mark 정보를 활용한다.(세부 내용은 ISO 14906 및 KS X ISO 14906 참조)		

### 6.5.2 INITIALIZATION.Request



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	8X (hex) 1000 x3	INITIALIZATION.Request x3.....선택 응용서비스 (사용 안 됨): 0
안테나 ID	x2x1 x0 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx	안테나 ID를 크게 두 개의 옥텟으로 나누어, 43비트 중 최상위 비트 쪽의 16 비트는 안테나 제조 ID, 나머지 27비트는 안테나 고유 ID를 나타냄 - 모두 MSB 퍼스트(First) 방식으로 전송
시간	XXXXXXXX (hex)	32 비트(범위 0~232-1)의 시간코드 - MSB 퍼스트(First) 방식으로 전송
프로파일	XX	응용 서비스를 제공하기 위한 물리적 성능 및 특성에 대한 정의 (프로파일 항목)
응용서비스 목록:		
필수 응용 서비스 개수	dd (dec)	지원 가능한 응용서비스의 수 - 현재는 ETC 기능만을 수행하므로 “1”
응용 서비스ID (AID)	x7x6000001 (bin)	지원 가능한 응용서비스의 ID - x7…0 (현재 EID 없음) - x6…0 (현재 파라미터 없음)
프로파일 목록:		
프로파일 개수 <sup>주3)</sup>	0	장비에 따라 프로파일 항목 선택 - 현재는 기본장비만 있으므로 “0”
주3)향후 확장성을 대비하여 단말기의 성능 및 특성에 따라 관리하는 관리번호의 개념으로 매체방식에 따라 다르게 부여한다. 예를 들어, 현재 사용되고 있는 ETC장비를 기본장비라 할 때, 이 장비의 프로파일 값은 “0”이 되고 향후 성능이나 특성이 개선된 장비가 출시되었을 때 프로파일 값은 순차적으로 증가하게 된다. 다만, 기본장비의 기능은 모든 장비가 수용하도록 하여야 한다. (세부내용은 ISO 15628(KS X ISO 15628) 및 KS 6915 참조)		

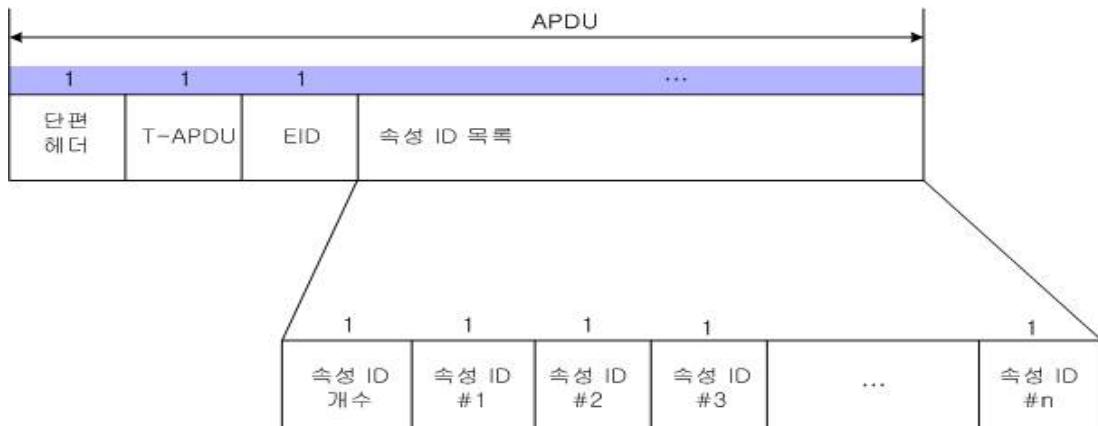
### 6.5.3 INITIALIZATION.Response



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001 (bin)	단편 정보
T-APDU	9X (hex) 1001 x3x2x1x0	INITIALIZATION.Response x3x2x1x0.....채움 비트 (사용 안 됨): 0000
프로파일	XX	응용 서비스 관련 프로파일 항목
응용서비스 목록:		
필수 응용 서비스 개수	01 (hex)	지원 가능한 응용 서비스의 수 - 현재는 ETC 기능만을 수행하므로 “1”
응용 서비스ID (AID) <sup>주4)</sup>	x7x6000001 (bin)	지원 가능한 응용서비스의 ID - x7…1 (현재 EID 있음) - x6…1 (현재 파라미터 있음)
EID	XX (hex)	OBU에 할당된 고유 EID
파라미터	Container	어플리케이션 관련 파라미터는 Context Mark <sup>주5)</sup> 를 포함한 하나의 Container 형식으로 전달 한다. <sup>주6)</sup>
OBU 구성	x7	현재 상태 정보 x7…1(상태정보 있음)
	xxxxxx xxxxxxxx (bin)	OBU 차종정보
	XX XX (hex)	제조자 ID
	x7x6x5x4x3x2x1x0 (bin)	x0 … 사용안함 x1 … 사용안함 x2 … 건전지 교체(1=저전압) x3 … ICC 없음(1=실패) x4 … ICC 불량(1=실패) x5 … 예외 S/W (1=예외) x6 … 지원 안 되는 H/W (사용안함) x7 … 영구적인H/W 불량(1=영구 불량)
	0000000x0 (bin)	x0 … 사용 안 함(향후 보완용)
주4) AID는 OBU가 제공 가능한 서비스를 나타내는 것으로 현재는 ETC서비스만 가능하므로 ETC에 해당하는 “1”을 적용한다. 향후 화물차량관리, 교통정보제공 등의 서비스로 확장할 경우에는 해당 서비스의 AID 번호를 추가하도록 하며, 이에 대한 원칙은 국토교통부에서 별도로 정한다. (세부내용은 ISO 15628(KS X ISO 15628) 참조)		
주5) 컨텍스트 마크는 같은 EID를 이용하더라도 계약상대가 다를 경우 각각에 대한 정보형태 등이 다를 수 있기 때문에 부여하는 정보이다. (이의 부여 및 관리에 대한 사항은 본 기준 5페이지, 세부방법 및 절차는 KS X ISO 14906 및 15628을 참조)		
※ 현재는 한국도로공사(하이패스플러스카드)만이 계약상대이기 때문에 계약처리의 단순화를 위하여 “0”값으로 처리 후 어플리케이션 파라미터를 추가한다.		
주6) 어플리케이션 관련 파라미터 전송은 ApplicationContextMark 내에 ETC-ContextMark와 어플리케이션 파라미터를 아래 표와 같이 추가한다. (ISO 14906.2004, p83, Annex B.4.2.4 Initialization Response(VST) 참조)		

범위	실제 값	설명	
파라미터 타입	0x02 (hex)	타입 2 : 옥텟 문자열	
파라미터 길이	0x06+Container1+Container2 +.....(hex)	ETC-ContextMark(6Byte) + n(어플리케이션 파라미터 총 길이)	
파라미터 값 Octet String	ETC-ContextMark	0xXXXXXXXXXXXXX	EFC 컨텍스트 마크
	Type	0x02	OCTET STRING
	Container Length	0xXX	어플리케이션 파라미터 #n1 길이
	Container Value	.....	어플리케이션 파라미터 #n1 값
	Type	0x02	OCTET STRING
	Container Length	0xXX	어플리케이션 파라미터 #n1 길이
	Container Value	.....	어플리케이션 파라미터 #n1 값

#### 6.5.4 Get.Request



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	6X (hex) 0110 x3x2x1 x0	GET.Request x3.....접근 권리...0(사용 안 됨) x2.....iid(invokerID, 호출자 ID)...0(사용 안 됨) x1.....속성 ID 목록...1(속성 ID 목록 있음) x0.....채움 비트...0(사용 안 됨)
EID	XX (hex)	Element ID
속성 ID 목록:		
속성 ID 개수	XX (hex)	현재 포함하고 있는 속성 ID의 개수
속성 ID 1	dd (dec)	첫 번째 속성 ID
속성 ID 2	dd (dec)	두 번째 속성 ID
속성 ID n	:	-

### 6.5.5 Get.Response



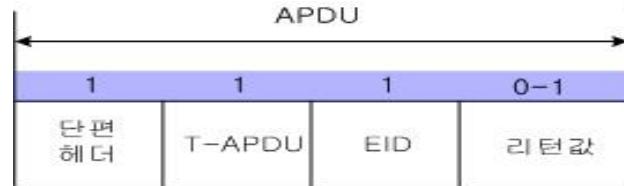
범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	7X (hex) 0111 x3x2x1 x0	<p>GET.Response            x3.....iid(invokerID, 호출자 ID)...0(사용 안됨)            x2.....속성 목록 :</p> <ul style="list-style-type: none"> <li>0 = 속성 목록 없음</li> <li>1 = 속성 목록 있음 (EID가 있는 경우)</li> </ul> <p>x1..... 리턴값 :</p> <ul style="list-style-type: none"> <li>0 = 리턴값 없음</li> <li>1 = 리턴값 있음</li> </ul> <p>x0.....채움 비트...0(사용 안 됨)</p>
EID	XX (hex)	Element ID
속성 목록:		
속성 개수	XX (hex)	현재 포함하고 있는 속성의 개수
속성 ID	dd (dec)	속성 ID
속성 타입	dd (dec)	타입
속성 길이	dd (dec)	길이
속성 내용	:	현재 속성의 내용
리턴 값	XX	리턴(상태) 값 (Optional)

### 6.5.6 Set.Request



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	4X (hex) 0100 x3x2x1 x0	SET.Request x3....접근 권리...0(사용 안 됨) x2....iid(invokerID, 호출자 ID)...0(사용 안됨) x1....모드: 0 = 비확인 1 = 확인 x0....채움 비트...0(사용 안 됨)
EID	XX (hex)	Element ID
속성 목록:		
속성 개수	XX (hex)	현재 포함하고 있는 속성의 개수
속성 ID	dd (dec)	속성 ID
속성 타입	dd (dec)	타입
속성 길이	dd (dec)	길이
속성 내용	:	현재 속성의 내용

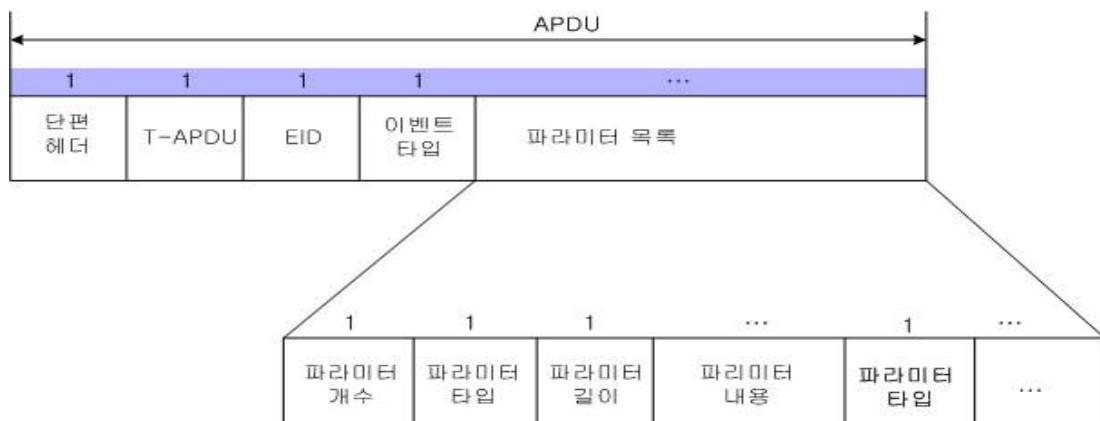
### 6.5.7 Set.Response



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	5X (hex) 0101 x3x2x1 x0	SET.Response x3.....iid(invokerID, 호출자 ID)…0(사용 안됨) x2.....리턴값 : 0 = 리턴값 없음 1 = 리턴값 있음 x1.....채움 비트…0(사용 안 됨) x0.....채움 비트…0(사용 안 됨)
EID	XX (hex)	Element ID
리턴 값	-	리턴(상태) 값 (Optional)

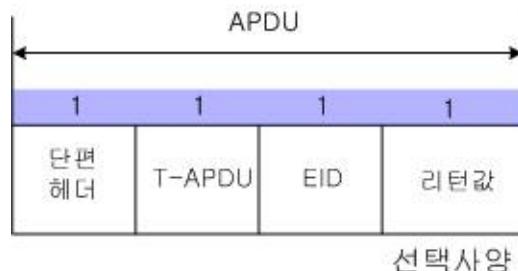
### 6.5.8 EVENT-REPORT.Request

- EVENT-REPORT.Request 서비스는 일반적인 메시지를 교환하거나 응용 서비스 단에 특별 이벤트를 알리는데 사용됨



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	2X (hex) 0010 x3x2x1 x0	<p>EVENT-REPORT.Request</p> <p>x3.....접근 권리…0(사용 안 됨)</p> <p>x2.....이벤트 파라미터:</p> <ul style="list-style-type: none"> <li>0 = 파라미터 없음</li> <li>1 = 파라미터 있음</li> </ul> <p>x1.....iid(invokerID, 호출자 ID)…0(사용 안 됨)</p> <p>x0.....모드:</p> <ul style="list-style-type: none"> <li>0 = 비확인</li> <li>1 = 확인</li> </ul>
EID	XX (hex)	Element ID
이벤트 타입	XX	응용 서비스에 종속된 1바이트 값
파라미터 목록:		
파라미터 개수	nn	현재 포함하고 있는 파라미터의 개수
파라미터 타입	-	응용 서비스에 종속된 값
파라미터 길이	-	응용 서비스에 종속된 값
파라미터 내용	-	응용 서비스에 종속된 값

### 6.5.9 EVENT-REPORT.Response



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	3X (hex) 0011 x3x2x1 x0	<p>EVENT-REPORT.Response</p> <p>x3..... iid(invokerID, 호출자 ID)…0(사용 안 됨)</p> <p>x2..... 리턴값:</p> <ul style="list-style-type: none"> <li>0 = 리턴값 없음</li> <li>1 = 리턴값 있음</li> </ul> <p>x1.....채움 비트…0(사용 안 됨)</p> <p>x0.....채움 비트…0(사용 안 됨)</p>
EID	XX (hex)	Element ID
리턴 값	XX	리턴(상태) 값 (선택 사양)

### 6.5.10 ACTION.Request



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	0X (hex) 0000 x3x2x1 x0	ACTION.Request x3.....접근 권리 : 0 = 사용안됨 1 = 사용됨 x2.....액션 파라미터: 0 = 파라미터 없음 1 = 파라미터 있음 x1.....iid(invokerID, 호출자 ID)...0(사용 안됨) x0.....모드: 0 = 비확인 1 = 확인
EID	XX (hex)	Element ID
옵션 타입	해당 응용 서비스에 따른 값	
액션 파라미터:		
파라미터 타입	-	해당 응용 서비스에 따른 값 (Container)
파라미터 길이	-	해당 응용 서비스에 따른 값
파라미터 내용	-	해당 응용 서비스에 따른 값

### 6.5.11 ACTION.Response



범위	실제 값	설명
단편 헤더	1x6x5x4 x3001(bin)	단편 정보
T-APDU	1X (hex) 0001 x3x2x1 x0	<p>ACTION.Response</p> <p>x3..... iid(invokerID, 호출자 ID)…0(사용 안됨)</p> <p>x2..... 응답 파라미터:</p> <p>0 = 파라미터 없음 1 = 파라미터 있음</p> <p>x1.....리턴값:</p> <p>0 = 리턴값 없음 1 = 리턴값 있음</p> <p>x0.....채움 비트…0(사용 안 됨)</p>
EID	XX (hex)	Element ID
응답 파라미터:		
파라미터 타입	-	해당 응용 서비스에 따른 값 (Container)
파라미터 길이	-	해당 응용 서비스에 따른 값
파라미터 내용	-	해당 응용 서비스에 따른 값
리턴 값	XX	리턴(상태) 값

### 6.5.12 리턴값

- OBU가 RSE에 정보를 전달하거나 RSE에 요구하는 명령을 처리하는 과정에서 발생된 결과에 대한 상태를 알려주기 위한 코드로 다음과 같은 값으로 표현됨

실제 값(hex)	설명	비고
00h	-	정상
01h	접근 불가	보안 관련한 문제로 인하여 접근이 불가능할 경우에 결과값
02h	인수 에러	해당 서비스에서 지원하지 않는 속성값이 발생했을 경우에 결과값
03h	복잡도 제한	여러 파라미터를 전달 또는 처리하는 과정에서 적어도 하나의 파라미터에서 에러가 발생했을 경우에 결과값
04h	처리 실패	명령의 처리 중 오류가 발생했을 경우에 결과값
05h	처리 오류	명령처리를 완료했으나 처리결과가 가용하지 않을 경우에 결과값
06h	연쇄화 실패	연속된 프레임정보를 분해 또는 조립하는 과정에서 오류가 발생했을 경우에 결과값

### 6.5.13 Context Mark(EID) 구성

- Context Mark의 구성은 계약에 주어진 서비스 권한을 발행한 기관을 식별하기 위한 ContractProvider, 계약에 적용되는 규칙에 대한 ContractProvider 특성을 표시하는 TypeOfContract, 주어진 ContractProvider의 상황 내에서 당해 계약의 구현 버전을 나타내는 ContextVersion으로 구성되어 있음

<표 31> Context Mark(EID) 구성

구분	노드ID 체계			범위
	형식		CountryCode STRING(SIZE(10)) BIT AA..ZZ	
ContractProvider	Provider	IssuerIdentifier INTEGER(0..16383)	0..16383	
TypeOfContract	OCTET STRING(SIZE(2))			
ContextVersion	INTEGER(0..127,...)			

- ContractProvider는 국가 코드인 10bit의 CountryCode와 발행 식별자인 14bit의 IssuerIdentifier로 구성되어 있음
  - CountryCode는 ISO 3166-1을 참조
  - IssuerIdentifier는 ISO 14819:2000의 Annex A를 참조

# 제2장

## 기술기준 적용 이해

### 1. ASN.1 범용툴을 이용한 기술기준 적용 방법

#### 1.1 ASN.1 범용툴 개요

##### 1.1.1 개발 배경

- ITS 표준은 시스템 간 정보교환에 적용하기 위해 표준화된 데이터 표현방식을 정의할 필요가 있으며, 국토교통부에서 제정·고시한 기술기준은 국제표준(ISO)의 기술적 동향에 따라 플랫폼 및 운영체제에 독립적인 방법으로 정보를 표현할 수 있는 추상표기구문인 ASN.1(Abstract Syntax Notation One)을 적용함
- 그러나 사업시행자는 ASN.1으로 정의된 기술기준을 적용하여 시스템을 개별적으로 구축하기에는 시간과 기술이 부족하고, 동일기술에 대한 중복투자로 경제적 손실이 발생함
- 국외는 민간기업에서 기계언어로의 변환을 위한 상용툴을 개발하여 사용되고 있으나 라이센스 비용, 툴 유지비용 등으로 구축 및 유지보수에 과다 비용이 발생함
- 이에 기술기준 적용의 용이성 및 교통정보 호환성 확보를 위해 정부차원에서 검증된 기술기준 구현 프로그램인 ASN.1 범용툴을 개발·제공하여 지원함
  - ASN.1 범용툴 개발로 인해 기 구축된 시스템 또는 향후 구축될 시스템은 별도의 추가적인 투자비용 없이 보다 편리하게 기술기준 적용 가능

##### 1.1.2 ASN.1 범용툴 기능

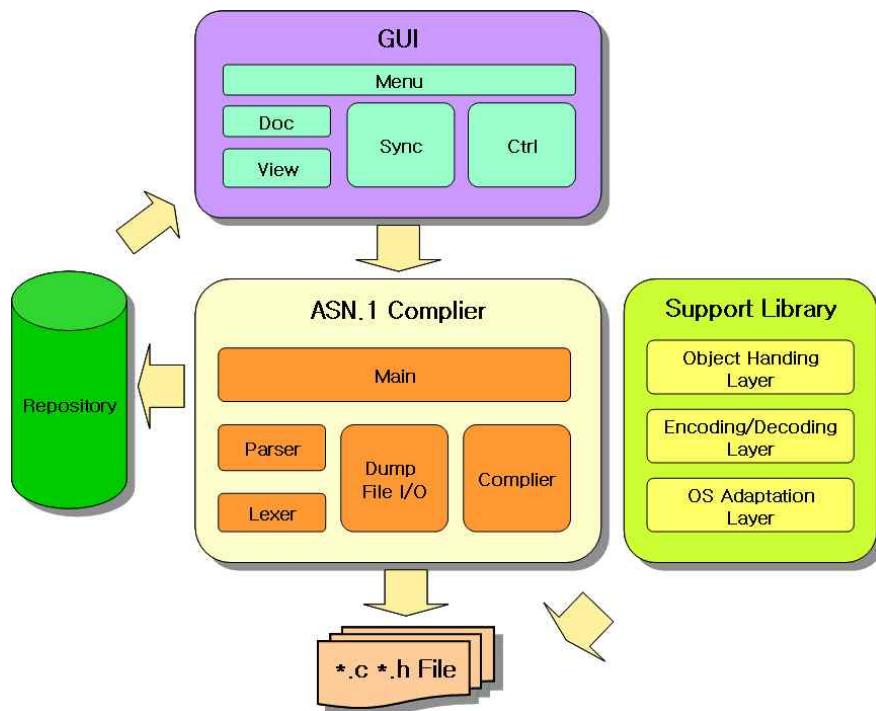
- ASN.1 범용툴의 기본 기능은 ASN.1으로 정의된 기술기준을 기존의 개발환경에서 사용할 수 있도록 이를 컨버팅 해주는 것이며 기술기준을 Window 기반의 C 언어로 구현하기 위해 범용툴에서 제공하는 기능은 다음과 같음

<표 32> ASN.1 범용툴 기능

기본 기능	내용
사용자 편의성 제공	<ul style="list-style-type: none"> <li>사용의 편의성을 제공하기 위하여 GUI 기반의 사용자 인터페이스를 제공하며 사용자는 GUI 인터페이스를 통하여 ASN.1 정의파일 선택 및 컴파일 가능</li> <li>ASN.1에 의하여 정의된 정보의 세부 정의 파악 가능</li> </ul>
C언어 코드 생성	<ul style="list-style-type: none"> <li>ASN.1 표준에 의하여 정의된 기술기준을 시스템에 적용할 수 있도록 C 언어 기반의 코드 생성</li> </ul>
전송구분 변환 기능	<ul style="list-style-type: none"> <li>생성된 정보를 타 센터로 전송하기 위하여 ASN.1 구문을 전송구문으로 변환 할 수 있는 기능 제공</li> </ul>
시스템의 효율성 확보	<ul style="list-style-type: none"> <li>기술기준 수정·변경 시 매번 기술기준 문서 전체에 대한 컴파일 작업을 수행 함으로써 자원 낭비</li> <li>변경된 ASN.1 정의만을 컴파일 할 수 있는 기능을 추가하여 시스템의 효율성 확보 가능</li> </ul>

### 1.1.3 ASN.1 범용툴 시스템 아키텍처

- ASN.1 범용툴은 표준에 정의된 교통정보를 ITS 시스템에 적용될 수 있는 C언어 코드로 생성해주는 기능을 수행하며, 범용툴의 기능 블록은 ASN.1 컴파일러, Repository, Support Library, GUI 등 네 가지로 구성됨



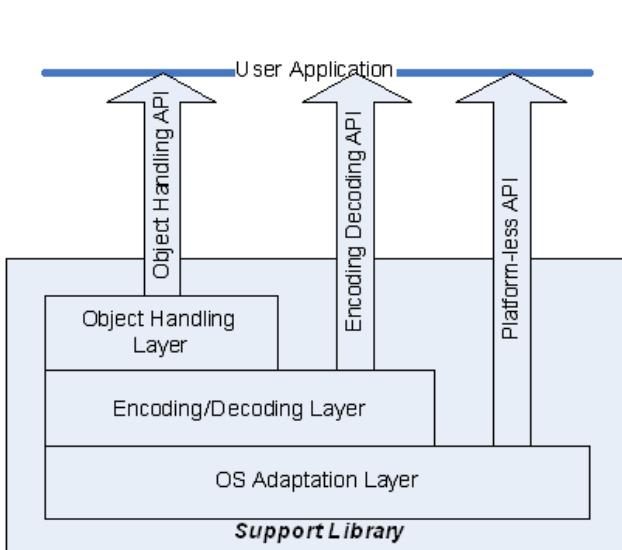
<그림 28> ASN.1 범용툴 시스템 아키텍처 구성도

## (가) ASN.1 컴파일러

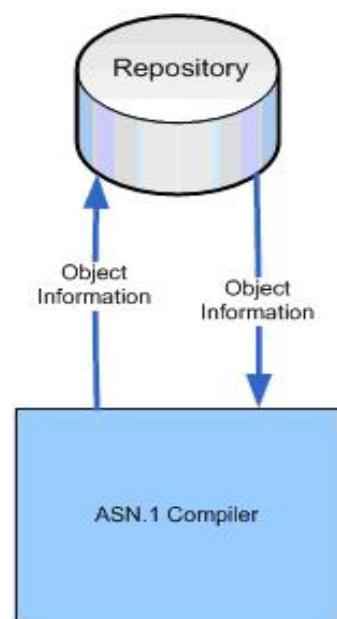
- ASN.1 컴파일러는 다음의 기능 블록으로 구성됨
  - Main 블록 : ASN.1 문서의 문법검사, 소스 및 헤더파일 생성을 위한 관리기능을 수행
  - Parser 블록 : ASN.1 문서의 문법 검사
  - Lexer 블록 : 문법검사를 위한 토큰 생성
  - Dump File I/O 블록 : Repository를 위한 Dump 파일 생성
  - Complier 블록 : 소스 및 헤더파일 생성

## (나) Support Library

- Support Library는 ASN.1 컴파일러에 의하여 생성된 C개발언어 소스 코드와 헤더파일이 정상적으로 동작할 수 있도록 지원함
  - Object Handling Layer : 기능 호출 시 메모리 생성 및 삭제
  - Encoding/Decoding Layer : Encoding/Decoding 기능 제공
  - OS Adaptation Layer : Windows, Unix, Linux의 시스템 운영체제 지원을 위한 기능 제공



<그림 29> Support Library Layer 구조



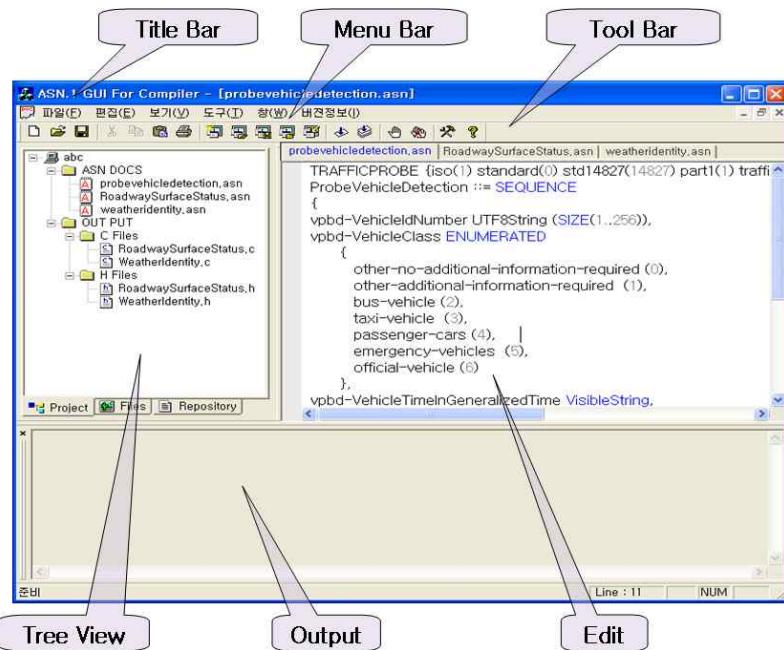
<그림 30> Repository 구조도

### (다) Repository

- 기존 상용툴에는 없는 기능으로 물리적인 파일저장 공간인 Repository를 이용하여 컴파일 된 문서의 재 컴파일 수행 시 빠른 처리를 위한 기능을 수행할 수 있음
- 사용자는 컴파일 된 문서의 자료구조를 Repository에 저장되어있는 Dump File을 이용하여 확인 할 수 있음
  - GUI로 ASN.1 Repository에 저장된 ASN.1 문서의 구조 확인 기능 제공
  - Repository 기능 제공을 위한 Dump File 저장
  - Dump File 기능을 제공함으로써 재 컴파일 시 이미 만들어진 Dump File을 이용하여 즉각적인 소스 및 헤더파일 생성 기능 제공
  - 컴파일 시간 단축

### (라) GUI

- GUI는 ASN.1 문서의 컴파일 시 편의기능을 제공하며 문서의 프로젝트 단위 관리 기능, 문서 편집 및 저장 등의 기능을 제공
  - ASN.1 문서의 작성, 편집, 저장 기능
  - 프로젝트 단위로의 생성, 저장 기능
  - 트리뷰를 통한 ASN.1 문서 검색 기능
  - 프로젝트뷰를 통한 프로젝트 단위의 관리 기능
  - 단일 컴파일기능 및 프로젝트 단위의 컴파일 기능
  - 컴파일 옵션 저장 기능
  - 컴파일된 ASN.1 문서의 자료구조를 보기 위한 Repository 화면 기능
  - 문서편집기에서 제공되는 일반적인 파일 edit(저장, 편집, 삭제 등) 기능



〈그림 31〉 ASN.1 GUI 화면 구성

- 위와 같은 기능을 수행하는데 사용되는 ASN.1 범용틀의 메뉴는 다음과 같음

〈표 33〉 범용틀 메뉴 기능

구분	메뉴	내용	메뉴	내용
파일	새 파일	새 문서	인쇄설정	프린터 설정
	열기	저장 문서 열기	새 프로젝트 생성	새 프로젝트 만들기
	닫기	열린 문서 닫기	프로젝트 열기	저장된 새 프로젝트 열기
	모두 닫기	열린 문서 모두 닫기	프로젝트 닫기	열린 새 프로젝트 닫기
	저장	작업 문서 저장하기	프로젝트 저장	작업 새 프로젝트 저장
	다른 이름으로 저장	다른 이름으로 저장하기	프로젝트 파일에 추가	프로젝트 파일에 추가하기
	인쇄	인쇄	종료	프로그램 종료
	인쇄 미리보기	인쇄 여백 설정		
편집	취소	입력 취소	다음 찾기	문장 중 다음 내용 검색
	잘라내기	작업 내용 잘라내기	바꾸기	입력한 내용으로 바꾸기
	복사	복사하기	북마크 표시/비표시	원하는 위치 표시 기능
	붙여넣기	복사 또는 잘라낸 내용 붙이기	북마크 삭제	표시한 부분 삭제
	찾기	입력한 내용 검색		
보기	폰트 변경	사용 폰트 변경	결과창	결과창 표시/비 표시
	트리뷰	트리뷰 표시/비 표시	도구모음	도구모음 표시/비 표시
도구	설정	컴파일 옵션 지정	프로젝트 컴파일	프로젝트 단위의 컴파일
	컴파일 실행	열린 문서의 컴파일	자동 줄 바꿈	자동 툴 바꿈

구분	메뉴	내용	메뉴	내용
창	다음 창	다음 창 보기	바둑판 식	바둑판 모양으로 보기
	이전 창	이전 창 보기	아이콘 정렬	아이콘 정렬
	계단식	계단식으로 보기		
버전정보	버전정보	프로그램의 버전 정보		

## 1.2 ASN.1 범용툴 설치

### 1.2.1 범용툴 획득

#### 1) 배포 정책

- 국토교통부에서는 기술기준을 적용하는 ITS 사업이나 ITS 표준 연구에 적극 활용될 수 있도록 ASN.1 범용툴의 배포 및 매뉴얼을 제공함
- 기술기준 적용을 위해 개발된 목적으로만 사용되도록 기타의 개인 또는 단체의 영리를 위한 목적으로 무분별한 배포 및 복제를 방지하기 위하여 허가된 시스템 또는 사용자에게만 사용될 수 있도록 관리중임
- ASN.1 범용툴은 국가ITS데이터등록소(<http://dr.its.go.kr>)에서 배포중이며, 웹사이트 이용이 제한된 경우, 표준화전담기관을 통해 받을 수 있음
- 국가의 ITS 관련 사업을 수행하는 경우 1년의 라이센스를 발급받아 ASN.1 범용툴을 사용 가능함



<그림 32> 국가 ITS 데이터등록소를 활용한 범용툴 배포 방법

## 2) 배포 현황

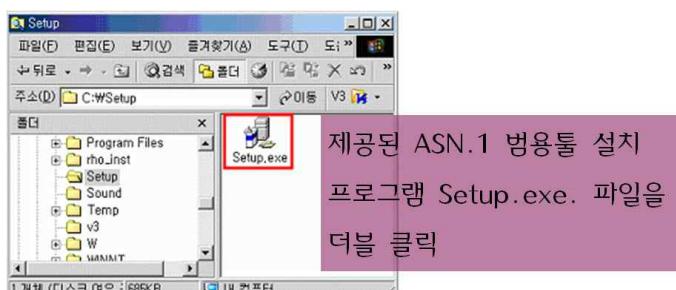
- ASN.1 범용툴을 배포하기 시작한 2007년부터 현재까지 약 80건의 ITS 사업에 범용툴을 배포하였음

<표 34> 사업별 ASN.1 범용툴 배포현황(2015년 8월 기준)

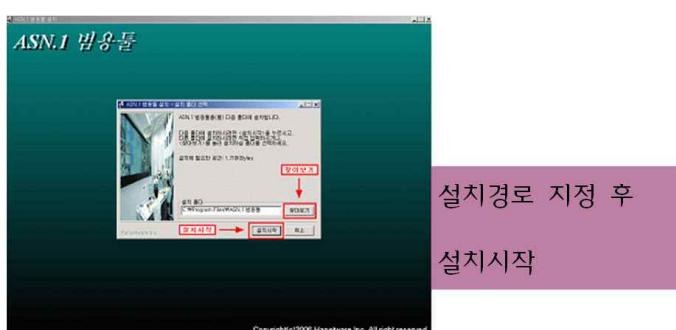
구분	건수	구분	건수
교통정보연계 사업(센터 등)	8	지방자치단체 ITS 사업	24
광역권 BIS 사업	20	고속도로 ITS 사업	6
연구개발 사업	7	교육	1
지방자치단체 BIS 사업	10	광역 민간 ITS 사업	2
국가사업(국가재난관리사업)	2	기타	-

### 1.2.2 범용툴 설치과정

#### ① 설치파일 실행



#### ② 설치경로 지정



#### ③ 설치결과 확인



### 1.2.3 범용틀 라이센스 발급

① key.bin 파일 생성

```
C:\WIN\라이센스 설치 데모>dir
C:\ 드라이브의 폴더에는 이름이 없습니다.
폴더 일련 번호: 447h-DEB2

C:\WIN\라이센스 설치 데모>make_key
C:\ 드라이브의 폴더에는 이름이 없습니다.
폴더 일련 번호: 447h-DEB2

C:\WIN\라이센스 설치 데모>dir
C:\ WIN\라이센스 설치 데모>make_key
C:\ 드라이브의 폴더에는 이름이 없습니다.
폴더 일련 번호: 447h-DEB2

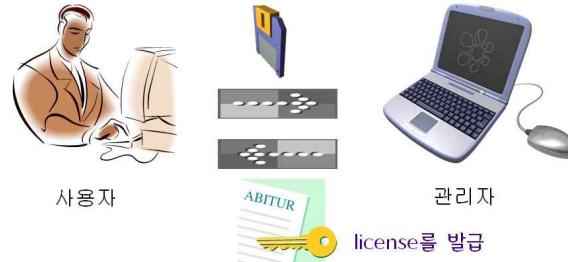
C:\WIN\라이센스 설치 데모>dir
2006-05-19 오전 10:28 <DIR> .
2006-05-19 오전 10:28 <DIR> ..
2006-05-18 오후 06:34 1,956,558 asnic.exe
2006-05-17 오후 05:34 633,454 install_license.exe
2006-05-18 오후 06:33 204 license.dat
2006-05-17 오후 04:48 245,812 make_key.exe
4개 파일 2,836,080 바이트 네임
2개 디렉터리 14,122,946,568 바이트 네임

C:\WIN\라이센스 설치 데모>make_key
C:\ 드라이브의 폴더에는 이름이 없습니다.
폴더 일련 번호: 447h-DEB2

C:\WIN\라이센스 설치 데모>dir
2006-05-19 오전 10:28 <DIR> .
2006-05-19 오전 10:28 <DIR> ..
2006-05-18 오후 06:34 1,956,558 asnic.exe
2006-05-17 오후 05:34 633,454 install_license.exe
2006-05-19 오후 10:28 204 key.bin
2006-05-18 오후 06:33 204 license.dat
2006-05-17 오후 04:48 245,812 make_key.exe
5개 파일 2,836,080 바이트 네임
2개 디렉터리 14,122,946,568 바이트 네임
```

② 생성된 key.bin 파일 관리자에게

전송 및 라이센스 수령



③ 라이센스 적용

```
C:\WIN\라이센스 설치 데모>install_license asnic.exe license.dat
License User ID : testuser
License User ID : testpass
licensing at 0x0002eac0
```

④ 설치경로의 license\_server파일을 System32(Win32) 폴더로 이동

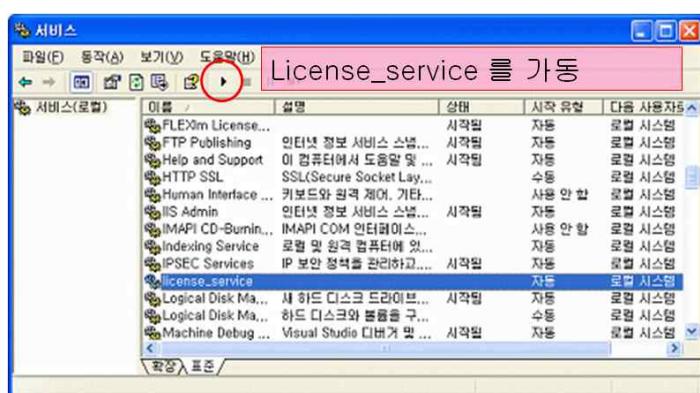
⑤ license\_server 가동

```
C:\WINDOWS\system32>license_server -ins
Installing Service ...
OpenSCManager OK
CreateService succeeded
Service installed OK

C:\WINDOWS\system32>
```

⑥ 시작>제어판>관리도구>서비스

실행

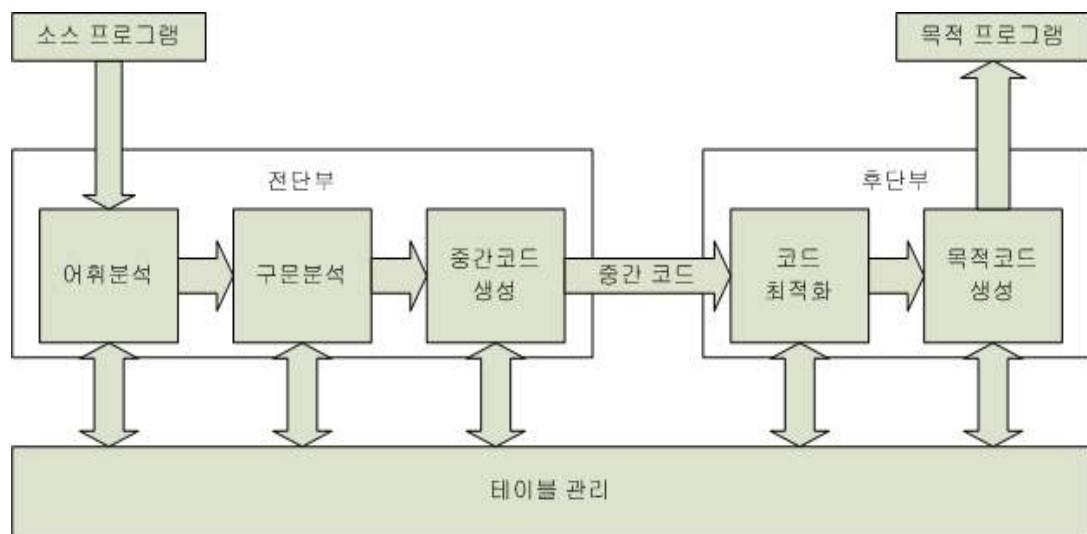


## 1.3 ASN.1 범용틀 기본 사용방법

### 1.3.1 ASN.1 컴파일 생성

#### 1) 컴파일러

- 컴파일러는 일정한 문법을 가진 형식의 구문을 정해진 표준에 의해서 검사하여 실행 가능한 형태로 번역해주는 프로그램을 말하고 컴파일러 구조는 크게 전단부(Front-end)와 후단부(Back-end)로 나눌 수 있음
- 전단부는 소스언어에 관계되는 부분으로써 소스 프로그램을 분석하고 어휘 분석 단계로부터 중간코드를 생성하는 역할을 담당함
- 후단부는 전단부에서 생성된 중간코드를 특정 기계를 위한 목적코드로 번역하는 기능을 수행하여 최종적으로 목적 프로그램을 생성함
- 일반적인 컴파일러의 구조는 다음과 같이 구성됨

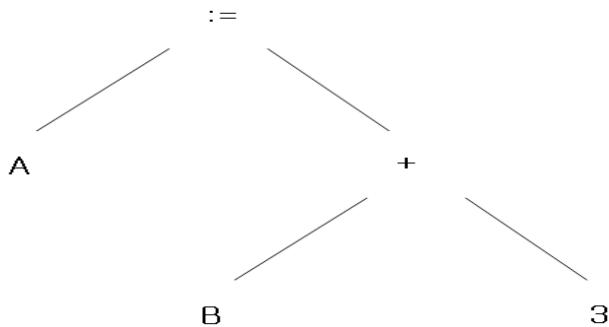


〈그림 33〉 컴파일러의 일반 구조

- 어휘분석 : 컴파일 과정의 첫 번째 단계로써 어휘분석기에 의해서 이루어지며 이는 소스 프로그램을 읽어 들여 일련의 토큰<sup>1)</sup>(Token)을 생성하도록 구성
- 구문분석 : 어휘분석 단계에서 생성된 토큰을 받아서 소스 프로그램에 대한 에러를 검사하고 올바른 문장에 대해서 구문구조 생성. 소스 프로그램에 에러가 있으면 그에 해당하는 에러 메시지를 출력하고 그렇지 않다면 프로그램에 대한 구문

1) 토큰이란 문법적 의미를 갖는 최소단위의 프로그램으로써 토큰의 열로 구성

구조를 트리 형태로 만들어 출력하도록 구성하고 트리 형태로는 파스트리와 (Parser Tree)와 추상 구문 트리(Abstract syntax tree)가 있으며 근래 대부분의 컴파일러에서는 추상 구문 트리 사용



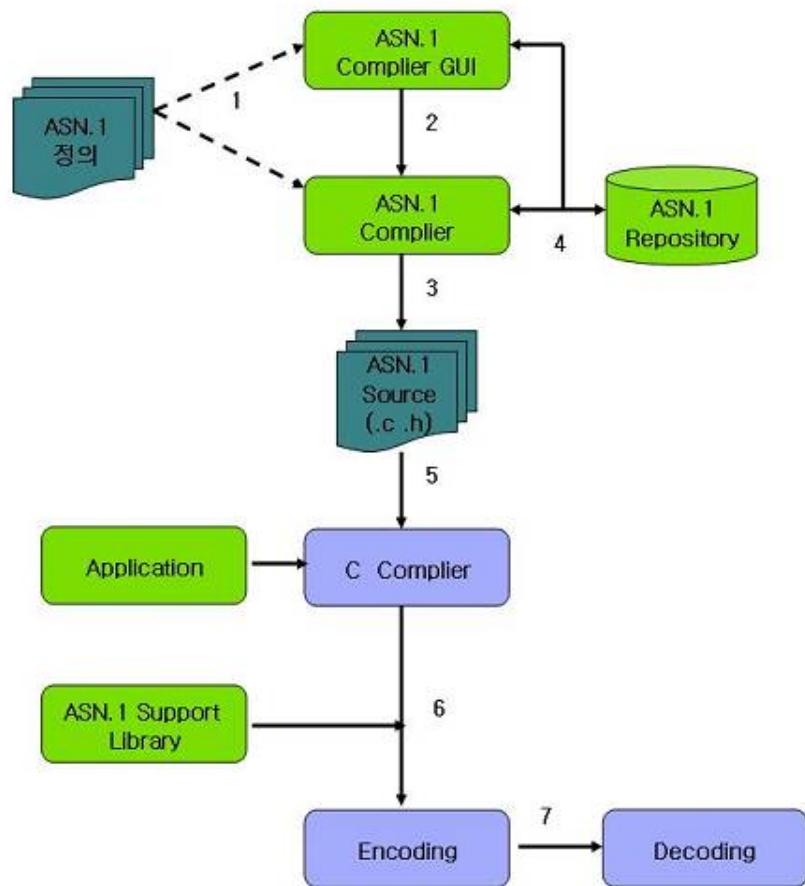
<그림 34> A := B+3;의 추상 구문 트리

- 중간코드 생성 : 파서의 출력인 추상 구문 트리의 입력을 받아서 의미 검사를 수행하고 그에 해당하는 중간 코드를 생성하는 것으로, 구문분석 단계에서 만들어진 구문구조를 이용하여 코드를 생성하고 구문 트리를 확인하여 각 구문에 대한 중간코드를 생성
- 코드 최적화 : 제공된 중간 코드를 의미적으로 동등하면서 좀 더 효율적인 코드로 변경하는 것으로 계산 횟수를 줄이고, 보다 빠른 명령을 사용하여 실행시간이 짧은 코드를 생성하며 메모리 사용을 최소화 하도록 구성
- 목적코드 생성 : 중간 코드를 입력받아서 그와 의미적으로 동등한 목적 기계에 대한 코드 생성

## 2) ASN.1 범용틀 컴파일 절차

- ASN.1로 정의된 교통정보를 ASN.1 범용틀을 사용하여 컴파일 하는 과정은 아래 그림과 같은 절차에 의해서 이루어짐
  - ① ASN.1에 의하여 정의된 기술기준을 ASN.1 GUI에서 또는 ASN.1 컴파일러 명령어 상에서 컴파일
  - ② ASN.1 컴파일러 GUI를 이용하는 경우에, 사용자가 컴파일 수행 버튼을 누르면 ASN.1 정의 파일 이름, 필요 옵션(사용자 메뉴얼 참조)등과 함께 ASN.1 컴파일러에 전달

- ③ ASN.1 컴파일러는 ASN.1 정의 파일의 내용을 분석(Parsing)하고 유효 여부를 검사한 후, C언어 형태의 소스 및 헤더 파일 생성
- ④ ASN.1 컴파일러는 기술기준 내용을 분석 한 후 해당 ASN.1 정의를 GUI 또는 기타 다른 용도로 사용될 수 있는 형식으로 ASN.1 Repository에 저장
- ⑤ ASN.1 컴파일러로부터 생성된 소스 및 헤더파일을 기 개발된 Application과 연동
- ⑥ 소스 및 헤더파일을 Support Library의 지원 기능과 함께 C 컴파일러를 통해 컴파일하여 전송대상 시스템에 연결(Link)해서 사용할 수 있는 PDU 생성
- ⑦ 생성된 PDU를 전송 대상 시스템에 전송하며 전송받은 PDU를 Decoding하여 전송된 데이터 추출



<그림 35> ASN.1 컴파일 과정

### 3) 런타임 라이브러리

- Support Library는 ASN.1 컴파일러에 의하여 생성된 C개발언어 소스 코드와 헤더 파일이 정상적으로 동작할 수 있도록 하는 지원 기능을 제공함
  - BER Encoding 규칙을 지원하는 Library 기능 제공
  - 센터에 설치된 각종 OS를 지원하는 Library 기능 제공
  - 기존 상용툴과의 연동기능 제공
  - 센터-센터의 파일 전송을 위해 생성되는 PDU 대한 지원기능 제공
- ASN.1 컴파일러에 의하여 생성된 C개발언어 소스 및 헤더파일이 정상적으로 동작할 수 있도록 하는 기본 지원 기능을 제공함
- Support Library는 데이터타입의 생성, 삭제, 프린트, Encoding, Decoding의 기능을 제공하며, 이는 기능별로 각기 다른 역할을 담당하는 여러 함수로 구성됨
- ASN.1에 의하여 정의된 모든 데이터에 관하여 다음과 같은 API 함수를 생성함

〈표 35〉 Support Library 생성 API

기본 API	설명
Encode_xxxx	C 언어 자료 구조에 저장된 데이터를 BER 네트워크로 전송할 수 있는 형태로 Encoding하는 API
Decode_xxxx	네트워크로부터 수신된 데이터를 BER로 Decoding하여 C 언어 프로그램에서 사용될 수 있는 C 언어 자료 구조로 저장하는 API
Print_xxxx	디버깅 또는 테스트를 위하여 C 언어 자료 구조에 저장된 데이터를 Print하는 API

#### 1.3.2 Type Descriptor

##### 1) 컴파일러 사용방법

- File Name : ASN.1 범용 Support Library는 제공 기능별로 각기 다른 역할을 담당하는 여러 함수들로 구성되며, 이는 개념적인 의미임
- 사용법 : asn1c [options | filenames]

사용법
asn1c 컴파일러는 ITU-T X.680-0207 표준 규격에 명시한 Syntax로 작성된 문서를 Compile하며, 입력 파일은 4개의 스테이지에 따라 처리 : Parsing, Semantic Check, Dump, Compile
asn1c 컴파일러는 입력 파일의 확장자를 이용하여 파일의 타입 구분
*.asn : ASN.1 소스 파일 *.asn.dump : ASN.1 Object Dump 파일
asn1c 컴파일러는 입력 파일을 컴파일 하여 출력 파일 생성
*.c : C 소스 파일 *.h : C 헤더 파일
옵션(반드시 띄어 써야 함)
‘-f’옵션과 -W옵션은 -f<name>과 같은 형태로 작성
-E ASN.1문서에 대하여 Parsing만 수행하고 생성된 Object Tree를 화면에 출력
-F -E 옵션과 같이 사용하며, Parsing후 구문 검사를 수행
-P ASN.1 문서를 Parsing후 인식된 ASN.1 Text를 화면에 출력
-X 화면 출력을 XML Format으로 출력
-D ASN.1 문서를 Parsing후 생성된 Object Tree를 Dump File로 변환
-Werror Compile시 발생 되는 경고를 에러로 취급
-Wdebug_lexer Lexer Block에서 Debug Message를 출력
-Wdebug_parser Parser Block에서 Debug Message를 출력
-Wdebug_fixer 구문 검사를 처리하는 과정에서 Debug Message를 출력
-Wdebug_compiler Compile Block에서 Debug Message를 출력
-fall-defs-global C 소스에서 “asn1_DEF_*”로 생성되는 구조체 인스턴스를 전역 변수로 선언
-fbless-SIZE C 소스에서 SIZE 제약 사항을 걸 수 없는 Data Type에도 SIZE제약 사항을 걸 수 있도록 허용
-fcompound-names 생성되는 Structure의 이름을 모두 명확히 기술
-findirect-choice CHOICE Object를 생성할 때, 값 대신 Pointer 참조
-fknown-extern-type=<name> ASN.1 문서를 Parsing할 때 지정한 name을 이미 알고 있는 type으로 동작
-fnative-types 변수 선언 시 가능한 INTEGER_t 대신 long 사용
-fno-constraints Object에 대한 제약사항을 검사하는 코드 비생성

-fno-include-deps	#include 구문 비생성
-funnamed-unions	구조체 내부에 이름이 지정되지 않은 union 사용 가능
-ftypes88	ASN.1 문서를 Parsing할 때 1988년 규격 타입만 지원
-print-constraint	-E옵션을 사용하여 ASN.1 Object Tree를 출력 할 때, Constraint에 대한 주석 출력
-print-lines	-E옵션을 사용하여 ASN.1 Object Tree를 출력 할 때, ASN.1 Object가 선언된 ASN.1 문서의 줄 번호를 주석으로 출력
-v	asn1c의 Version 출력
-h	Help Message 출력

## 2) 런타임 라이브러리

- Support Library는 기본적인 User API를 Type Descriptor를 통해서 제공함
- OCTET STRING을 디코딩하고자 하는 경우, 사용자 Source Code에서 “asn\_DEF\_OCTET\_STRING.ber\_decoder” 를 호출하며 Type Descriptor의 구조는 다음과 같음

Structure Name	asn_TYPE_descriptor_t	
Member		
Type	Name	Description
char *	name	Type의 이름
char *	xml_tag	XML Encode시 사용할 태그 명
asn_struct_free_f *	free_struct	해당 Type을 Free하는 함수의 주소
asn_struct_print_f *	print_struct	해당 Type의 내용을 출력하는 함수의 주소
asn_constr_check_f *	check_constraints	해당 Type의 constraint를 검사하는 함수의 주소
ber_type_decoder_f *	ber_decoder	주어진 Buffer를 BER에 의해 Decode하여 해당 Type을 만들어 내는 함수의 주소
der_type_encoder_f *	der_encoder	해당 Type을 DER에 의해 Encode하여 Data Stream을 만들어 내는 함수의 주소
다음 Member들은 내부적으로 처리하기 위해 정의하고, User Application에서 다음에 열거한 Member Function이나 Member 변수를 사용하여서는 안 됨		
asn_outmost_tag_f *	outmost_tag	해당 Type의 TAG를 추출하는 함수(Optional, Internal)
ber_tlv_tag_t *	tags	Sequence, Set, Choice에서 유효한 Member에 대한 Tag값을 저장하는 배열

int	tags_count	tags의 개수
ber_tlv_tag_t *	all_tags	Sequence, Set, Choice에서 모든 Member에 대한 Tag값을 저장하는 배열
int	all_tags_count	all_tags의 개수
asn_TYPE_member_s *	elements	Sequence, Set, Choice의 Member에 대한 정보를 저장하는 배열
int	elements_count	elements의 개수
void *	specifics	해당 Type만의 고유한 정보를 저장하고 있는 Structure의 주소

## (가) free\_struct

- free\_struct member Function은 메모리에 할당된 특정 Type Structure를 해제하는 함수임

Prototype	void free_struct(struct asn_TYPE_descriptor_s *type_descriptor, void *struct_ptr, int free_contents_only)
Description	메모리에 할당된 struct_ptr를 해제하는 것으로, free_contents_only가 0이 아닌 값으로 설정되면, struct_ptr의 내용만 Memory에서 해제하고 struct_ptr은 해제하지 않음
Return Value	없음
Arguments	
Name	Description
type_descriptor	해당 Type에 대한 정보를 담고 있는 Type Descriptor의 주소
struct_ptr	Memory에서 해제할 Data Buffer의 주소
free_contents_only	해제 방법을 설정하는 Flag
Example	
<pre>#include &lt;asn_system.h&gt; #include &lt;OCTET_STRING.h&gt; static int text_output(const void *buffer, size_t size, void *param) {     unsigned int i;     FILE *fp = (FILE *)param;     unsigned char *cp = (unsigned char *)buffer;     for (i = 0; i &lt; size; i++) fputc((int)*cp++, fp);     fputc('\n', fp);     return size; } int main(void) {     char *string = "test string";     OCTET_STRING_t *octetString = </pre>	

```

        OCTET_STRING_new_fromBuf(&asn_DEF_OCTET_STRING,
                                string,
                                strlen(string));
    asn_DEF_OCTET_STRING.print_struct(&asn_DEF_OCTET_STRING,
                                       octetString,
                                       0,
                                       text_output, (void *)stdout);
    asn_DEF_OCTET_STRING.free_struct(&asn_DEF_OCTET_STRING,
                                      octetString,
                                      0);
    return 0;
}

```

#### (4) print\_struct

- print\_struct은 해당 Type Structure의 내용을 사람이 읽을 수 있는 형태로 화면에 출력해 주는 함수임

Prototype	<code>int print_struct(struct asn_TYPE_descriptor_s *type_descriptor, const void *struct_ptr, int level, asn_app_consume_bytes_f *callback, void *app_key)</code>
Description	type_descriptor에 의거하여 struct_ptr의 내용을 사람이 읽을 수 있는 형태로 변환하여 화면에 출력
Return Value	성공 시 0을 실패 시 -1을 돌려줌
Arguments	
Name	Description
type_descriptor	해당 Type에 대한 정보를 담고 있는 Type Descriptor의 주소
struct_ptr	화면에 출력할 Data Buffer의 주소
level	들여쓰기 수준 값
call_back	사용자가 정의한 Data를 화면에 출력할 때 사용할 사용자 출력함수의 주소
app_key	call_back에 지정된 함수가 수행하는데 필요한 인수를 저장한 Memory의 주소
Example	

```

#include <asn_system.h>
#include <OCTET_STRING.h>
static int text_output(const void *buffer, size_t size, void *param)
{
    unsigned int i;
    FILE *fp = (FILE *)param;
    unsigned char *cp = (unsigned char *)buffer;
    for (i = 0; i < size; i++) fputc((int)*cp++, fp); fputc('\n', fp);
    return size;
}
int main(void)
{

```

```

    char *string = "test string";
    OCTET_STRING_t *octetString =
OCTET_STRING_new_fromBuf(&asn_DEF_OCTET_STRING,
                        string, strlen(string));
    asn_DEF_OCTET_STRING.print_struct(&asn_DEF_OCTET_STRING, octetString,
                                      0, text_output, (void *)stdout);
    return 0;
}

```

#### (다) check\_constraints

- check\_constraints는 해당 Type Structure의 내용이 ASN.1 문서에서 정의한 규칙에 위배되는 것이 없는지 검사해 주는 함수임

Prototype	int check_constraints (struct asn_TYPE_descriptor_s *type_descriptor, const void *struct_ptr, asn_app_consume_bytes_f *optional_app_errlog, void *optional_app_key);
Description	struct_ptr의 내용이 ASN.1 문서에서 정의한 규칙에 위배하는 것이 없는지 검사
Return Value	성공 시 0을 실패 시 -1을 돌려줌
Arguments	
Name	Description
type_descriptor	해당 Type에 대한 정보를 담고 있는 Type Descriptor의 주소
struct_ptr	규칙 검사를 수행할 Data Buffer의 주소
optional_app_errlog	에러 발생시 에러 메시지를 처리할 사용자 함수의 주소
optional_app_key	optional_app_errlog를 수행하는데 필요한 정보를 저장한 Memory의 주소
Example	
<pre>#include &lt;asn_system.h&gt; #include &lt;OCTET_STRING.h&gt; static int text_output(const void *buffer, size_t size, void *param) {     unsigned int i;     FILE *fp = (FILE *)param;     unsigned char *cp = (unsigned char *)buffer;     for (i = 0; i &lt; size; i++) fputc((int)*cp++, fp); fputc('\n', fp);     return size; } int main(void) {     char *string = "test string";     OCTET_STRING_t *octetString = OCTET_STRING_new_fromBuf(&amp;asn_DEF_OCTET_STRING, string, strlen(string));     asn_DEF_OCTET_STRING.check_constraints(&amp;asn_DEF_OCTET_STRING,   octetString,   text_output, (void *)stdout);     return 0; }</pre>	

## (라) ber\_decoder

- ber\_decoder는 주어진 Buffer를 BER에 의해 디코딩하여 해당 Type의 구조를 생성해 주는 함수임

Prototype	asn_dec_rval_t ber_decoder(struct asn_codec_ctx_s *opt_codec_ctx, struct asn_TYPE_descriptor_s *type_descriptor, void **struct_ptr, const void *buf_ptr, size_t size, int tag_mode);
Description	buf_ptr에 주어진 Data를 Decoding하여 type_descriptor에 대응하는 Structure를 생성
Return Value	asn_dec_rval_t를 돌려줌
Arguments	
Name	Description
opt_codec_ctx	Decoding에 사용되는 Stack Memory의 양을 설정하는 Structure의 주소
type_descriptor	해당 Type에 대한 정보를 담고 있는 Type Descriptor의 주소
struct_ptr	생성된 Structure의 주소를 저장할 변수
buf_ptr	BER에 의해 Encoding된 Data가 저장되어 있는 Buffer의 주소
size	buf_ptr의 길이
tag_mode	Tag처리를 위한 Flag. -1인 경우 IMPLICIT Tag Mode, 0인 경우 No Tag Mode, 1인 경우 EXPLICIT Tag Mode로 동작
Example	
<pre>#include &lt;asn_system.h&gt; #include &lt;OCTET_STRING.h&gt; static unsigned char *load_file(char *fn, int *size) {     unsigned char *buf = NULL;     FILE *fp = fopen(fn, "r+b");     fseek(fp, 0, SEEK_END); *size = ftell(fp); fseek(fp, 0, SEEK_SET);     buf = (unsigned char *)malloc(*size);     fread(buf, *size, 1, fp); fclose(fp);     return buf; } int main(void) {     unsigned char *buf = NULL;     int size = 0;     buf = load_file("octetString.bin", &amp;size);     asn_DEF_OCTET_STRING.ber_decoder(NULL, &amp;asn_DEF_OCTET_STRING, &amp;result, buf, size, 1);     free(buf);     return 0; }</pre>	

- asn\_dec\_rval\_t structure는 다음과 같음

Structure Name		
Member		
Type	Name	Description
asn_dec_rval_code_e	code	결과 Code
size_t	consumed	Decoding에 사용된 Buffer의 Byte수

- code에는 다음과 같은 값이 저장됨

Name	Value
RC_OK	0
RC_WMORE	1
RC_FAIL	2

- Decoding 후 code가 RC\_WMORE인 경우 Decoding에 필요한 Data가 충분하지 않다는 것을 의미함

### (마) der\_encoder

- der\_encoder는 해당 Type의 구조를 주어진 DER에 의해 인코딩해 주는 함수임

Prototype	asn_enc_rval_t der_encoder(struct asn_TYPE_descriptor_s *type_descriptor, void *struct_ptr, int tag_mode, ber_tlv_tag_t tag, asn_app_consume_bytes_f *consume_bytes_cb, void *app_key)
Description	struct_ptr에 저장된 Data를 BER에 의해 Encoding
Return Value	asn_enc_rval_t structure를 돌려줌
Arguments	
Name	Description
type_descriptor	해당 Type에 대한 정보를 담고 있는 Type Descriptor의 주소
struct_ptr	Encoding할 Data가 저장된 Data Buffer의 주소
tag_mode	Tag처리를 위한 Flag. -1인 경우 IMPLICIT Tag Mode, 0인 경우 No Tag Mode, 1인 경우 EXPLICIT Tag Mode로 동작
consume_bytes_cb	Encoding된 Data를 처리할 사용자 함수의 주소
app_key	consume_bytes_cb에 지정된 함수가 수행하는데 필요한 인수를 저장한 Memory의 주소
Example	
<pre>#include &lt;asn_system.h&gt; #include &lt;OCTET_STRING.h&gt; static int save_output(const void *buffer, size_t size, void *param) {     FILE *fp = (FILE *)param;     fwrite(buffer, size, 1, fp);</pre>	

```

        return size;
    }
int main(void)
{
    char *string = "test string";
    asn_dec_rval_t ret;
    OCTET_STRING_t *octetString = OCTET_STRING_new_fromBuf(&asn_DEF_OCTET_STRING,
                                                             string, strlen(string));
    FILE *fp = fopen("octetString.bin", "wb");
    asn_DEF_OCTET_STRING.der_encoder(&asn_DEF_OCTET_STRING, octetString,
                                      1, asn_DEF_OCTET_STRING.tags[0],
                                      save_output, (void *)fp);
    fclose(fp);
    return 0;
}

```

### (B) der\_encoder function

- asn\_enc\_rval\_t는 다음과 같음

Structure Name		asn_enc_rval_t
Member		
Type	Name	Description
ssize_t	encoded	Encoding된 Data의 크기로, -1인 경우 Encoding에 실패한 것을 의미
asn_TYPE_descriptor_s *	failed_type	Encoding에 실패한 Data의 Type Descriptor 주소
void *	structure_ptr	Encoding에 실패한 Data의 주소

- Support Library가 기본적으로 제공하는 ASN Type에 대하여 Type Descriptor 가 하나씩 존재하며, 각 ASN Type별 Type Descriptor의 이름과 이것을 정의한 헤더는 다음과 같음

ASN Type	Type Descriptor Name	Header File
ANY	asn_DEF_ANY	ANY.h
BIT_STRING	asn_DEF_BIT_STRING	BIT_STRING.h
BOOLEAN	asn_DEF_BOOLEAN	BOOLEAN.h
ENUMERATED	asn_DEF_ENUMERATED	ENUMERATED.h
GeneralString	asn_DEF_GeneralString	GeneralString.h
GeneralizedTime	asn_DEF_GeneralizedTime	GeneralizedTime.h
GraphicString	asn_DEF_GraphicString	GraphicString.h
IA5String	asn_DEF_IA5String	IA5String.h

ASN Type	Type Descriptor Name	Header File
INTEGER	asn_DEF_INTEGER	INTEGER.h
ISO64String	asn_DEF_ISO64String	ISO64String.h
NULL	asn_DEF_NULL	NULL.h
NativeEnumerated	asn_DEF_NativeEnumerated	NativeEnumerated.h
NativeInteger	asn_DEF_NativeInteger	NativeInteger.h
NativeReal	asn_DEF_NativeReal	NativeReal.h
NumericString	asn_DEF_NumericString	NumericString.h
OBJECT_IDENTIFIER	asn_DEF_OBJECT_IDENTIFIER	OBJECT_IDENTIFIER.h
OCTET_STRING	asn_DEF_OCTET_STRING	OCTET_STRING.h
ObjectDescriptor	asn_DEF_ObjectDescriptor	ObjectDescriptor.h
PrintableString	asn_DEF_PrintableString	PrintableString.h
REAL	asn_DEF_REAL	REAL.h
RELATIVE-OID	asn_DEF_RELATIVE-OID	RELATIVE-OID.h
T61String	asn_DEF_T61String	T61String.h
TeletexString	asn_DEF_TeletexString	TeletexString.h
UTCTime	asn_DEF_UTCTime	UTCTime.h
UTF8String	asn_DEF_UTF8String	UTF8String.h
UniversalString	asn_DEF_UniversalString	UniversalString.h
VideotexString	asn_DEF_VideotexString	VideotexString.h
VisibleString	asn_DEF_VisibleString	VisibleString.h

### 1.3.3 Support Type 및 보조 함수

#### 1) Primitive Type

- Primitive Type은 Type Descriptor를 가지지 않아 직접 사용하지 않고 ASN.1에서 정의하는 Primitive Class Data Type을 정의하기 위한 기본 구조임
- Primitive Type 구조는 다음과 같음

Structure Name		ASN__PRIMITIVE_TYPE_t
Member		
Type	Name	Description
uint8_t *	buf	Data가 저장되는 Buffer
int	size	buf의 크기

- Primitive Type Data 구조의 상속 관계는 다음과 같음

- |                                       |
|---------------------------------------|
| Primitive Type – INTEGER – ENUMERATED |
| – OBJECT_IDENTIFIER – RELATIVE-OID    |
| – REAL                                |

## (가) INTEGER Type 보조 함수

Prototype	int asn_INTEGER2long(const INTEGER_t *i, long *l);
Description	INTEGER Type에 저장된 값을 long type 변수에 저장
Return Value	성공 시 0을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
i	INTEGER Type Structure의 주소
l	변환된 값이 저장될 Native Type 변수의 주소
Example	
<pre>int main(void) {     INTEGER_t i;     long initial = 100, data = 0;     int status = 0;     status = asn_long2INTEGER(&amp;i, initial);     status = asn_INTEGER2long(&amp;i, &amp;data); }</pre>	

Prototype	int asn_long2INTEGER(INTEGER_t *i, long l)
Description	long type에 저장된 값을 INTEGER Type Structure에 저장
Return Value	성공 시 0을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
i	변환된 값이 저장될 INTEGER Type Structure의 주소
l	값이 저장된 long type 변수
Example	
<pre>int main(void) {     INTEGER_t i;     long initial = 100, data = 0;     int status = 0;     status = asn_long2INTEGER(&amp;i, initial);     status = asn_INTEGER2long(&amp;i, &amp;data); }</pre>	

Prototype	const asn_INTEGER_enum_map_t *INTEGER_map_value2enum(asn_INTEGER_specifics_t *specs, long value)
Description	Enumerate로 정의한 INTEGER Type Structure의 정보를 이용하여, value에 해당하는 Enumerate 정보를 찾아냄
Return Value	해당 정보를 찾아내면 찾아낸 asn_INTEGER_enum_map_t structure의 주소를, 찾아내지 못하면 NULL을 돌려줌
Arguments	
Name	Description
specs	해당 Type에 대한 type descriptor의 specifics
value	Enumerate 정보를 찾을 숫자 값이 저장된 변수
Example	
<pre>int main(void) {     long enum_val = 3;     asn_INTEGER_enum_map_t *map = INTEGER_map_value2enum(         (asn_INTEGER_specific_t *)asn_DEF_INTEGER.specific, enum_val); }</pre>	

#### (4) OBJECT\_IDENTIFIER 보조 함수

Prototype	int OBJECT_IDENTIFIER_get_arcs(OBJECT_IDENTIFIER_t *_oid, void *_arcs, unsigned int _arc_type_size, unsigned int _arc_slots)
Description	Object Identifier Structure에서 arc값을 얻어옴
Return Value	성공 시 얻어낸 arc의 개수를, 실패 시 -1을 돌려줌
Arguments	
Name	Description
_oid	arc를 얻어올 Object Identifier Structure의 주소
_arcs	arc를 저장할 배열의 주소
_arc_type_size	_arcs의 단위 크기(예 : sizeof(_arcs[0]))
_arc_slots	_arcs배열의 원소 개수
Example	
<pre>void oid_example(OBJECT_IDENTIFIER_t *oid) {     long array[16];     int slot_size = sizeof(array[0]), i;     int status = OBJECT_IDENTIFIER_get_argc(oid, array,  slot_size, sizeof(array)/slot_size);     for (i = 0; i &lt; 16; i++)         printf("%ld\n", array[i]); }</pre>	

Prototype	<code>int OBJECT_IDENTIFIER_set_arcs(OBJECT_IDENTIFIER_t *_oid, void *_arcs, unsigned int _arc_type_size, unsigned int _arc_slots)</code>
Description	Object Identifier Structure에서 arc값 설정
Return Value	성공 시 설정한 arcs의 개수를, 실패 시 -1을 돌려줌
Arguments	
Name	Description
_oid	arc를 설정할 Object Identifier Structure의 주소
_arcs	설정할 arc를 저장하고 있는 배열
_arc_type_size	_arcs의 단위 크기(예 : sizeof(_arcs[0]))
_arc_slots	_arcs배열의 원소 개수
Example	
<pre>void oid_example(OBJECT_IDENTIFIER_t *oid) {     long arcs[] = { 1, 3, 6, 1, 4, 1 };     int slot_size = sizeof(arcs[0]);     int status = OBJECT_IDENTIFIER_set_arcs(oid, arcs,  slot_size, sizeof(arcs)/slot_size) }</pre>	

Prototype	<code>int OBJECT_IDENTIFIER_parse_arcs(const char *oid_text, ssize_t oid_txt_length, long arcs[], unsigned int arcs_slots, const char **opt_oid_text_end)</code>
Description	Text로 작성되어 있는 OID를 해석하여 arcs배열에 저장
Return Value	성공 시 해석한 OID의 arc개수를, 실패 시 -1을 돌려줌
Arguments	
Name	Description
oid_text	Text로 작성된 OID가 저장된 Buffer의 주소
oid_txt_length	oid_text의 길이
arcs[]	해석된 OID의 arc값이 저장될 배열
arcs_slots	arcs배열의 크기
opt_oid_text_end	해석 종료 후, 남은 Text의 시작 위치
Example	
<pre>void oid_example(void) {     char *text = "1.3.6.1.4.1";     long arcs[16] = { 0, };     int i;     int cnt = OBJECT_IDENTIFIER_parse_arcs(text, -1, arcs, 16, NULL);     for (i=0; i&lt;cnt; i++)         printf("%ld\n", arcs[i]); }</pre>	

Prototype	<code>int RELATIVE_OID_get_arcs(RELATIVE_OID_t *_roid, void *arcs, unsigned int arc_type_size, unsigned int arc_slots)</code>
Description	Relative OID Structure에서 arc값을 얻어옴
Return Value	성공 시 얻어낸 arc의 개수를, 실패 시 -1을 돌려줌
Arguments	
Name	Description
_roid	arc를 얻어올 Relative OID Structure의 주소
arcs	arc값을 저장할 배열
arc_type_size	arc배열 원소의 크기
arc_slots	arc배열 원소의 개수

Prototype	<code>int RELATIVE_OID_set_arcs(RELATIVE_OID_t *_roid, void *arcs, unsigned int arc_type_size, unsigned int arcs_slots)</code>
Description	Relative OID Structure에 arc값 설정
Return Value	성공 시 설정한 arc의 개수를, 실패 시 -1을 돌려줌
Arguments	
Name	Description
_roid	arc값을 설정할 Relative OID Structure의 주소
arcs	설정할 값을 저장하고 있는 arc배열
arc_type_size	arc배열 원소의 크기
arcs_slots	arc배열 원소의 개수

Prototype	<code>int asn_REAL2double(const REAL_t *real_ptr, double *d)</code>
Description	REAL Structure에 저장된 정보를 double 변수로 변환
Return Value	성공 시 0을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
real_ptr	변환할 REAL Structure의 주소
d	변환된 값이 저장될 변수의 주소

Prototype	<code>int asn_double2REAL(REAL_t *real_ptr, double d)</code>
Description	double 변수에 저장된 값을 변환하여 REAL Structure에 저장
Return Value	성공 시 0을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
real_ptr	변환된 값을 저장할 REAL Structure의 주소
d	변환할 값을 저장한 Double변수

## 2) Any Type

- Any Type은 모든 Type을 수용 가능하며 그 구조는 다음과 같음

Structure Name		ANY_t	
Member			
Type	Name	Description	
uint8_t *	buf	data가 저장되는 Buffer	
int	size	buf의 크기	
asn_struct_ctx_t	_asn_ctx	Data를 조작하기 위한 부가 정보	

### (가) ANY\_fromType

Prototype	int ANY_fromType(ANY_t *from, asn_TYPE_descriptor_t *td, void *struct_ptr)
Description	struct_ptr에 지정된 내용을 type descriptor의 정보를 이용하여 ANY Type Structure에 저장
Return Value	성공 시 0을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
from	변환된 Data가 저장될 ANY Structure의 주소
td	변환할 Data에 대한 Type Descriptor의 주소
struct_ptr	변환될 Data가 저장된 Structure의 주소
Example	
<pre>void any_example(void) {     char *string = "test_string";     ANY_t from;     OCTET_STRING_t *octetString = OCTET_STRING_new_fromBuf(         &amp;asn_DEF_OCTET_STRING, string, strlen(string));     ANY_fromType(&amp;from, &amp;asn_DEF_OCTET_STRING, octetString);     return ret; }</pre>	

### (나) ANY\_new\_fromType

Prototype	ANY_t *ANY_new_fromType(asn_TYPE_descriptor_t *td, void *struct_ptr)
Description	struct_ptr에 지정된 내용을 type descriptor의 정보를 이용하여 ANY Type Structure로 생성
Return Value	성공 시 생성된 ANY Structure의 주소를, 실패 시 NULL을 돌려줌
Arguments	
Name	Description
td	변환할 Data에 대한 Type Descriptor의 주소

struct_ptr	변환할 Data를 저장한 Structure의 주소
Example	
	<pre>ANY_t *octetString2Any(OCTET_STRING_t *octetString) {     ANY_t *ret = ANY_new_fromType(&amp;asn_DEF_OCTET_STRING, octetString);     return ret; }</pre>

#### (다) ANY\_to\_type

Prototype	int ANY_to_type(ANY_t *to, asn_TYPE_descriptor_t *td, void **struct_ptr)
Description	ANY Structure에 저장된 정보를 Type Descriptor를 이용하여 변환한 다음 해당 Type의 Structure를 생성하고 변환된 Data를 생성한 Structure에 저장
Return Value	성공 시 0을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
to	변환할 Data를 저장한 ANY Structure의 주소
td	변환될 Data에 대한 Type Descriptor의 주소
struct_ptr	생성된 Structure의 주소가 저장될 변수
Example	
	<pre>OCTET_STRING_t *any2OctetString(ANY_t *any) {     OCTET_STRING_t *octetString = NULL;     ANY_to_type(any, &amp;asn_DEF_OCTET_STRING, &amp;octetString);     return octetString; }</pre>

#### (라) ANY\_fromBuf

Prototype	ANY_fromBuf(s, buf, size)
Description	주어진 Buffer를 이용하여 ANY Structure를 설정하고 이 매크로는 ANY_fromType 함수 호출
Arguments	
Name	Description
s	ANY Structure의 주소
buf	Buffer의 주소
size	Buffer의 길이
Example	
	<pre>void any_example(void) {     ANY_t any;     unsigned char buf[128] = { 0xff, };     ANY_fromBuff(&amp;any, buf, 128); }</pre>

### (마) ANY\_new\_fromBuf

Prototype	ANY_new_fromBuf(buf, size)
Description	주어진 Buffer를 이용하여 ANY Structure를 생성하고 이 매크로로 ANY_new_fromType 함수 호출
Arguments	
Name	Description
buf	Buffer의 주소
size	Buffer의 길이
Example	
<pre>void any_example(void) {     unsigned char buf[128] = { 0xff, };     ANY_t *any = ANY_new_fromBuf(buf, 128); }</pre>	

### 3) BIT\_STRING Type

- BIT\_STRING Type은 Binary Data를 저장하는 Type이며 그 구조는 다음과 같음

Structure Name	BIT_STRING_t	
Member		
Type	Name	Description
uint8_t *	buf	data가 저장되는 Buffer
int	size	buf의 크기
int	bits_unused	buf의 마지막 Byte에서 사용하지 않는 Bit의 수
asn_struct_ctx_t	_asn_ctx	Data를 조작하기 위한 부가 정보

### 4) OCTET\_STRING Type

- OCTET\_STRING Type은 문자열 Data를 저장하는 Type이고 그 구조는 다음과 같음

Structure Name	OCTET_STRING_t	
Member		
Type	Name	Description
uint8_t *	buf	data가 저장되는 Buffer
int	size	buf의 크기
asn_struct_ctx_t	_asn_ctx	Data를 조작하기 위한 부가 정보

## (가) OCTET\_STRING\_fromBuf

Prototype	int OCTET_STRING_fromBuf(OCTET_STRING_t *s, const char *str, int size)
Description	주어진 Buffer를 이용하여 Octet String Structure를 설정
Return Value	성공 시 0을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
s	값을 설정 할 Octet String Structure의 주소
*str	buffer의 주소
size	buffer의 길이
Example	
<pre>void octet_string_example(void) {     unsigned char buf[128] = { 0xff, };     OCTET_STRING_t octetString;     OCTET_STRING_fromBuff(&amp;octetString, buf, 128); }</pre>	

## (나) OCTET\_STRING\_new\_fromBuf

Prototype	OCTET_STRING_t *OCTET_STRING_new_fromBuf(asn_TYPE_descriptor_t *td, const char *str, int size)
Description	주어진 Buffer를 이용하여 Octet String Structure를 생성
Return Value	성공 시 생성된 Octet String Structure의 주소를, 실패 시 NULL을 돌려줌
Arguments	
Name	Description
td	OCTET STRING Type에 대한 Type Descriptor의 주소
str	Buffer의 주소
size	Buffer의 길이
Example	
<pre>void octet_string_example(void) {     unsigned char buf[128] = { 0xff, };     OCTET_STRING_t *octetString = OCTET_STRING_fromBuff(buf, 128); }</pre>	

## (다) codeconv\_open

Prototype	codeconv_t codeconv_open(char *fromcs, char *tocs)
Description	fromcs에 주어진 Code Set을 tocs에 주어진 Code Set으로 변환하는 Code Converting Handle을 Open

Return Value	성공 시 생성된 Code Converter Handle의 주소를, 실패 시 NULL을 돌려줌
Arguments	
Name	Description
fromcs	원본 Text의 Code Set Name
tocs	변환하고자 하는 Code Set Name
Example	
<pre>void utf8String_example(void) {     codeconv_t handle = codeconv_open("UTF8", "EUCKR");     char *nativeStr = "한글KSC5601", *nsp = nativeStr;     char *utf8Str[128], *usp = utf8Str;     size_t size = codeconv(handle, &amp;nsp, NULL, &amp;usp, NULL);     codeconv_close(handle); }</pre>	

## (라) codeconv

Prototype	size_t codeconv(codeconv_t cd, char **inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft)
Description	Code Converter Handle을 이용하여 문자열 변환
Return Value	변환한 문자열의 개수를 돌려줌
Arguments	
Name	Description
cd	Code Converter Handle
inbuf	변환할 문자가 저장된 buffer pointer의 주소로 함수가 종료되면 다음 변환을 시작할 위치 지시
inbytesleft	함수가 종료되면 inbuf의 정보를 사용하고 남은 바이트 수 저장
outbuf	변환된 문자가 저장될 buffer pointer의 주소와 함수가 종료되면 다음 변환된 문자가 저장될 위치 지시
outbytesleft	함수가 종료되면 outbuf를 사용하고 남은 바이트 수 저장
Example	
<pre>void utf8String_example(void) {     codeconv_t handle = codeconv_open("UTF8", "EUCKR");     char *nativeStr = "한글KSC5601", *nsp = nativeStr;     char *utf8Str[128], *usp = utf8Str;     size_t size = codeconv(handle, &amp;nsp, NULL, &amp;usp, NULL);     codeconv_close(handle); }</pre>	

## (파) codeconv\_close

Prototype	int codeconv_closeonv_t cd)
Description	Code Converter Handle 닫음
Return Value	항상 0을 리턴
Arguments	
Name	Description
cd	Code Converter Handle
Example	
<pre>void utf8String_example(void) {     codeconv_t handle = codeconv_open("UTF8", "EUCKR");     char *nativeStr = "한글KSC5601", *nsp = nativeStr;     char *utf8Str[128], *usp = utf8Str;     size_t size = codeconv(handle, &amp;nsp, NULL, &amp;usp, NULL);     codeconv_close(handle); }</pre>	

## (파) asn\_strings\_get\_length

Prototype	asn_strings_get_length(x)
Description	OCTET STRING Type 또는 OCTET STRINT Type에서 생성된 모든 String 계열의 Type에 대한 String의 길이 추출
Arguments	
Name	Description
x	OCTET STRING계열의 Structure주소
Example	
<pre>void octet_string_example(void) {     unsigned char buffer[128] = { 0xff, }, *buf = NULL;     OCTET_STRING_t *octetString = OCTET_STRING_new_fromBuff(buffer, 128);     unsigned char *buf = malloc(asn_strings_get_length(octetString)); }</pre>	

## (사) OCTET-STRING\_fromString

Prototype	OCTET_STRING_fromString(s, str)
Description	String을 이용하여 OCTET STRING Structure를 설정하고 이 매크로로 OCTET_STRING_fromBuf 호출
Arguments	
Name	Description
s	String을 설정할 OCTET STRING Structure의 주소
str	문자열이 저장된 Buffer의 주소

#### Example

```
void octet_string_example(void)
{
    unsigned char *str = "test string";
    OCTET_STRING_t octetString;
    OCTET_STRING_fromString(&octetString, str);
}
```

## 5) GeneralizedTime Type

### (가) asn\_GT2time

Prototype	time_t asn_GT2time(const GeneralizedTime_t *gt, struct tm *_optional_tm4fill, int as_gmt)
Description	GeneralizedTime에 저장된 값을 time_t로 변환
Return Value	성공 시 변환된 초 값을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
gt	변환할 값을 저장하고 있는 GeneralizedTime Structure의 주소
_optionl_tm4fill	변환된 값을 저장할 tm structure의 주소(Optional)
as_gmt	_optionl_tm4fill structure에 저장할 시간을 local time으로 저장할지, GMT Time으로 저장할지 결정하는 flag. (_optionl_tm4fill이 NULL이 아닌 경우에만 유효)
Example	
void GT_example(GeneralizedTime_t *gt)	{ time_t jiffie = asn_GT2time(gt, NULL, 0); printf("time = %ld\n", jiffie); }

### (나) asn\_GT2time\_frac

Prototype	time_t asn_GT2time_frac(const GeneralizedTime_t *tp, int *frac_value, int *frac_digits, struct tm *_optional_tm4fill, int as_gmt);
Description	GeneralizedTime에 저장된 값을 time_t로 변환하는 것으로, frac_value 및 frac_digits 값이 주어지면 1초 이하의 값을 변환하여 frac_value 및 frac_digit에 저장
Return Value	성공 시 변환된 초 값을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
tp	변환할 값을 저장하고 있는 GeneralizedTime Structure의 주소
frac_value	1초 이하의 시간 값이 저장될 변수의 주소
frac_digits	1초 이하의 소수점 자릿수

_optional_tm4fill	변환된 값을 저장할 tm structure의 주소(Optional)
as_gmt	_optionl_tm4fill structure에 저장할 시간을 local time으로 저장할지, GMT Time으로 저장할지 결정하는 flag. (_optionl_tm4fill이 NULL이 아닌 경우에만 유효)
Example	
<pre>void GT_example(GeneralizedTime_t *gt) {     int value = 0, digit = 0;     time_t jiffie = asn_GT2time_frac(gt, &amp;value, &amp;digit, NULL, 0);     double t = (double)jiffie + value * (1 / pow(10, digit));     printf("time = %lf\n", t); }</pre>	

#### (다) asn\_GT2time\_prec

Prototype	time_t asn_GT2time_prec(const GeneralizedTime_t *tp, int *frac_value, int frac_digits, struct tm *_optional_tm4fill, int as_gmt)
Description	GeneralizedTime에 저장된 값을 time_t로 변환하는 것으로, frac_value 및 frac_digits가 값이 주어지면 1초 이하의 값을 변환하여 frac_value 및 frac_digit에 저장
Return Value	성공 시 변환된 초 값을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
tp	변환할 값을 저장하고 있는 GeneralizedTime Structure의 주소
frac_value	1초 이하의 시간 값이 저장될 변수의 주소
frac_digits	1초 이하의 소수점 자릿수
_optional_tm4fill	변환된 값을 저장할 tm structure의 주소(Optional)
As_gmt	_optionl_tm4fill structure에 저장할 시간을 local time으로 저장할지, GMT Time으로 저장할지 결정하는 flag. (_optionl_tm4fill이 NULL이 아닌 경우에만 유효)
Example	
<pre>void GT_example(GeneralizedTime_t *gt) {     int value = 0, digit = 3;     time_t jiffie = asn_GT2time_prec(gt, &amp;value, digit, NULL, 0);     double t = (double)jiffie + value * (1 / pow(10, digit));     printf("time = %.3lf\n", t); }</pre>	

#### (라) asn\_time2GT

Prototype	GeneralizedTime_t *asn_time2GT(GeneralizedTime_t *_optional_gt, const struct tm *tm, int force_gmt)
Description	tm structure의 내용을 변환하여 GeneralizedTime에 설정하고 _optional_gt가 NULL이면 GeneralizedTime 생성

Return Value	성공 시 설정된(혹은 생성된) GeneralizedTime Structure의 주소를, 실패 시 NULL을 돌려줌
Arguments	
Name	Description
_optional_gt	변환된 시간 값이 저장될 GeneralizedTime Structure의 주소(Optional)
tm	변환할 시간 값이 저장된 tm Structure의 주소
force_gmt	설정(혹은 생성)되는 GeneralizedTime Structure를 GMT로 설정하도록 하는 Flag
Example	
<pre>void GT_example(void) {     time_t now = time(NULL);     struct tm *tmp = localtime(&amp;now);     struct tm tm = *tmp;     GeneralizedTime_t *gt = asn_time2GT(NULL, &amp;tm, 0); }</pre>	

### (마) asn\_time2GT\_frac

Prototype	GeneralizedTime_t *asn_time2GT_frac(GeneralizedTime_t *_optional_gt, const struct tm *tm, int frac_value, int frac_digits, int force_gmt);
Description	tm structure의 내용을 변환하여 GeneralizedTime에 설정하고 _optional_gt가 NULL이면 GeneralizedTime 생성
Return Value	성공시 설정된(혹은 생성된) GeneralizedTime Structure의 주소를, 실패 시 NULL을 돌려줌
Arguments	
Name	Description
_optional_gt	변환된 시간 값이 저장될 GeneralizedTime Structure의 주소(Optional)
tm	변환할 시간 값이 저장된 tm Structure의 주소
frac_value	1초 이하의 시간 값
frac_digits	1초 이하 시간 값의 소수점 자릿수
force_gmt	설정(혹은 생성)되는 GeneralizedTime Structure를 GMT로 설정하도록 하는 Flag
Example	
<pre>void GT_example(void) {     struct timeval tNow = { 0, 0 };     struct tm *tmp = NULL, tm;     GeneralizedTime_t *gt;      gettimeofday(&amp;tNow, NULL);     tmp = localtime(&amp;tNow.tv_sec);     tm = *tmp;     gt = asn_time2GT_frec(NULL, &amp;tm, tNow.tv_nsec, 9, 0); }</pre>	

## (비) asn\_UT2time

Prototype	time_t asn_UT2time(const UTCTime_t *tp, struct tm *_optional_tm4fill, int as_gmt);
Description	UTCTime에 저장된 값을 time_t로 변환
Return Value	성공 시 변환된 초 값을, 실패 시 -1을 돌려줌
Arguments	
Name	Description
tp	시간 값이 저장된 UTCTime Structure의 주소
_optional_tm4fill	변환된 값을 저장할 tm structure의 주소(Optional)
as_gmt	_optionl_tm4fill structure에 저장할 시간을 local time으로 저장할지, GMT Time으로 저장할지 결정하는 flag. (_optionl_tm4fill이 NULL이 아닌 경우에만 유효)
Example	
<pre>void UT_example(UTCTime_t *ut) {     time_t jiffie = asn_UT2time(ut, NULL, 0);     printf("time = %ld\n", jiffie); }</pre>	

## (사) asn\_time2UT

Prototype	UTCTime_t *asn_time2UT(UTCTime_t *__opt_ut, const struct tm *tm, int force_gmt);
Description	tm structure의 내용을 변환하여 UTCTime에 설정하고 _optional_gt가 NULL이면 UTCTime을 생성
Return Value	성공시 설정된(혹은 생성된) UTCTime Structure의 주소를, 실패 시 NULL을 돌려줌
Arguments	
Name	Description
__opt_ut	변환된 값을 저장할 UTCTime Structure의 주소
tm	변환할 시간 값이 저장된 tm Structure의 주소
force_gmt	설정(혹은 생성)되는 UTCTime Structure를 GMT로 설정하는 Flag
Example	
<pre>void UT_example(void) {     time_t now = time(NULL);     struct tm *tmp = localtime(&amp;now);     struct tm tm = *tmp;     UTCTime_t *ut = asn_time2UT(NULL, &amp;tm, 0); }</pre>	

## 6) UTF8String Type

### (가) UTF8String\_length

Prototype	ssize_t UTF8String_length(const UTF8String_t *st);
Description	UTF8String에 저장된 문자열이 WCS로 변환되었을 때의 길이 계산
Return Value	성공 시 문자열의 길이를, 실패 시 다음과 같은 값을 돌려줌 - 1 : UTF8 시퀀스가 잘림 - 2 : UTF8 시퀀스 시작 값이 틀림 - 3 : 연속 예외 처리 실패 - 4 : 처리하기 위한 최소 길이보다 Data가 적음 - 5 : 잘못된 인수
Arguments	
Name	Description
st	문자열 길이를 계산할 정보가 저장된 UTF8String Structure의 주소
Example	
<pre>void utf8String_example(void) {     UTF8String_t *utf8 = NULL;     codeconv_t handle = codeconv_open("UTF8", "EUCKR");     char *nativeStr = "한글KSC5601";     char *utf8Str[128];     size_t size = codeconv(handle, nativeStr, NULL, utf8Str, NULL);     ssize_t test = 0;     codeconv_close(handle);     utf8 = (UTF8String_t *)OCTET_STRING_new_fromBuff(utf8Str, size);     test = UTF8String_length(utf8);     printf("Native = %d, UTF8 = %d\n", strlen(nativeStr), test); }</pre>	

### (나) UTF8String\_to\_wcs

Prototype	size_t UTF8String_to_wcs(const UTF8String_t *st, uint32_t *dst, size_t dstlen);
Description	UTF8String Structure에 저장된 Data를 WCS 문자열로 변환
Return Value	UTF8String_length의 Return Value 참조
Arguments	
Name	Description
st	변환할 Data가 저장된 UTF8String Structure의 주소
dst	변환된 Data가 저장될 Buffer의 주소
dstlen	Buffer의 길이
Example	
<pre>void utf8String_example(UTF8String *st) {     ssize_t size = UTF8String_length(st);     char *buf = malloc(size + 1);     UTF8String_to_wcs(st, buf, size); }</pre>	

### 1.3.4 범용틀을 이용한 컴파일 및 예제

#### 1) ASN.1 Document

- Rectangle이라는 Type을 다음과 같이 정의함

```
RectangleModule DEFINITION ::= BEGIN
    Rectangle ::= SEQUENCE {
        height      INTEGER,
        width       INTEGER
    }
END
```

#### 2) 컴파일

- 상기에서 작성한 Rectangle.asn을 컴파일하여 Rectangle.h와 Rectangle.c를 얻음

```
c:\asn1c> asn1c -fnative-types Rectangle.asn1
```

#### 3) 코드구성

- 다음 Source는 Rectangle.h에 정의된 asn\_DEF\_Rectangle Type Descriptor를 이용하여, Rectangle\_t Type의 구조를 인코딩함

```
#include <stdio.h>
#include <sys/types.h>
#include <Rectangle.h> /* Rectangle ASN.1 type */

static int
write_out(const void *buffer, size_t size, void *app_key) {
    FILE *out_fp = app_key;
    size_t wrote;

    wrote = fwrite(buffer, 1, size, out_fp);

    return (wrote == size) ? 0 : -1;
}

int main(int ac, char **av) {
    Rectangle_t *rectangle; /* Type to encode */
```

```

asn_enc_rval_t ec;                                /* Encoder return value */

/* Allocate the Rectangle_t */
rectangle = calloc(1, sizeof(Rectangle_t)); /* not malloc! */
if(!rectangle) {
    perror("calloc() failed");
    exit(71); /* better, EX_OSERR */
}

/* Initialize the Rectangle members */
rectangle->height = 42; /* any random value */
rectangle->width  = 23; /* any random value */

/* BER encode the data if filename is given */
if(ac < 2) {
    fprintf(stderr, "Specify filename for BER output\n");
} else {
    const char *filename = av[1];
    FILE *fp = fopen(filename, "wb"); /* for BER output */

    if(!fp) {
        perror(filename);
        exit(71); /* better, EX_OSERR */
    }

    /* Encode the Rectangle type as BER (DER) */
    ec = der_encode(&asn_DEF_Rectangle,
                    rectangle, write_out, fp);
    fclose(fp);
    if(ec.encoded == -1) {
        fprintf(stderr,
                "Could not encode Rectangle (at %s)\n",
                ec.failed_type ? ec.failed_type->name : "unknown");
        exit(65); /* better, EX_DATAERR */
    } else {
        fprintf(stderr, "Created %s with BER encoded Rectangle\n",
                filename);
    }
}

return 0; /* Encoding finished successfully */
}

```

- 다음 Source는 Rectangle.h에 정의된 asn\_DEF\_Rectangle Type Descriptor를 이용하여, Rectangle\_t Type의 구조를 디코딩함

```
#include <stdio.h>
#include <sys/types.h>
#include <Rectangle.h>                                /* Rectangle ASN.1 type */

int main(int ac, char **av) {
    char buf[1024];                                     /* Temporary buffer      */
    Rectangle_t *rectangle = 0;                          /* Type to decode */
    asn_dec_rval_t rval;                               /* Decoder return value */
    FILE *fp;                                         /* Input file handler   */
    size_t size;                                       /* Number of bytes read */
    char *filename;                                    /* Input file name */

    /* Require a single filename argument */
    if(ac != 2) {
        fprintf(stderr, "Usage: %s <file.ber>\n", av[0]);
        exit(64); /* better, EX_USAGE */
    } else {
        filename = av[1];
    }
    /* Open input file as read-only binary */
    fp = fopen(filename, "rb");
    if(!fp) {
        perror(filename);
        exit(66); /* better, EX_NOINPUT */
    }

    /* Read up to the buffer size */
    size = fread(buf, 1, sizeof(buf), fp);
    fclose(fp);
    if(!size) {
        fprintf(stderr, "%s: Empty or broken\n", filename);
        exit(65); /* better, EX_DATAERR */
    }

    /* Decode the input buffer as Rectangle type */
    rval = ber_decode(0, &asn_DEF_Rectangle,
                     (void **)&rectangle, buf, size);
    if(rval.code != RC_OK) {
        fprintf(stderr, "%s: Broken Rectangle encoding at byte %ld\n",
                filename, (long)rval.consumed);
        exit(65); /* better, EX_DATAERR */
    }

    return 0; /* Decoding finished successfully */
}
```

## 1.4 BIS 개발 시 ASN.1 범용틀 활용 및 사례

### 1.4.1 ASN.1의 이해

#### 1) ASN.1란

- ASN.1은 구조화된 정보를 기술하기 위한 언어로 추상 구문표기법(Abstract Syntax Notation One)의 약어임
  - ASN.1은 OSI의 분야에 있어서 데이터형을 정의하기 위해 사용하는 기술 언어
- BER과 DER은 ASN.1 형태의 추상 데이터 형식을 네트워크상에서 전송할 수 있는 형식으로 encoding해주고 그것을 다시 ASN.1 형태로 decoding해주는 방법에 대하여 정의함
  - 네트워크 상에서 교환되는 메시지는 ASN.1형태로 구성되어 교환되지만, ASN.1 자체로는 추상적인 데이터 형식이므로 그것을 그대로 전송할 수 없음

#### 2) ASN.1의 필요성

- 각기 다른 시스템(예: UNIX와 WINDOWS)마다 데이터 표기법이 다르기 때문에 데이터를 전송해도 수신하는 쪽과 송신하는 쪽의 데이터가 다르게 해석될 수 있음
  - 즉, Windows는 리틀엔디언(Little Endian)이어서 

4	3	2	1
---	---	---	---

, 오른쪽부터 해석되고, UNIX계열은 빅엔디언(Big Endian)이어서 

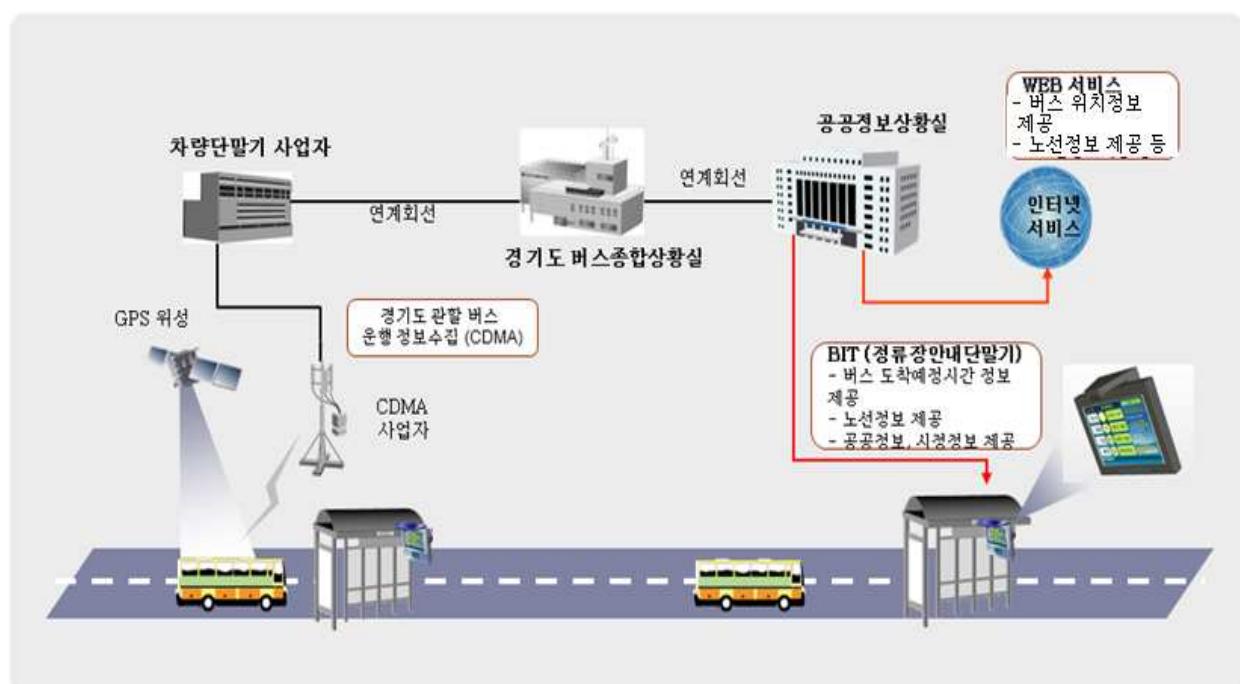
1	2	3	4
---	---	---	---

, 왼쪽부터 해석되는 방식이기 때문에 서로 다른 데이터로 인식함  
예) 0x23을 32비트를 처리하는 빅엔디언에서는 (0x23 0x00 0x00 0x00)로 표기 되지만 리틀엔디언에서는 (0x00 0x00 0x00 0x23)으로 표기
- 또한 운영하는 장비가 8비트, 16비트, 32비트, 64비트 중 어느 것인지에 따라 변수의 크기가 달라지고, 송수신 시스템 사이에 동일한 언어를 사용하지 않고 다른 언어를 사용한다면 더욱 복잡해지게 됨  
예) 32비트 장비에서 long형은 4바이트이고, 64비트 장비에서 long형은 8바이트 임
- 이러한 문제를 해결하기 위해서 공통화된 데이터 표현문법을 만들게 되었고 이를 통해서 ASN.1이 제정되었음

## 1.4.2 어플리케이션 개발 활용

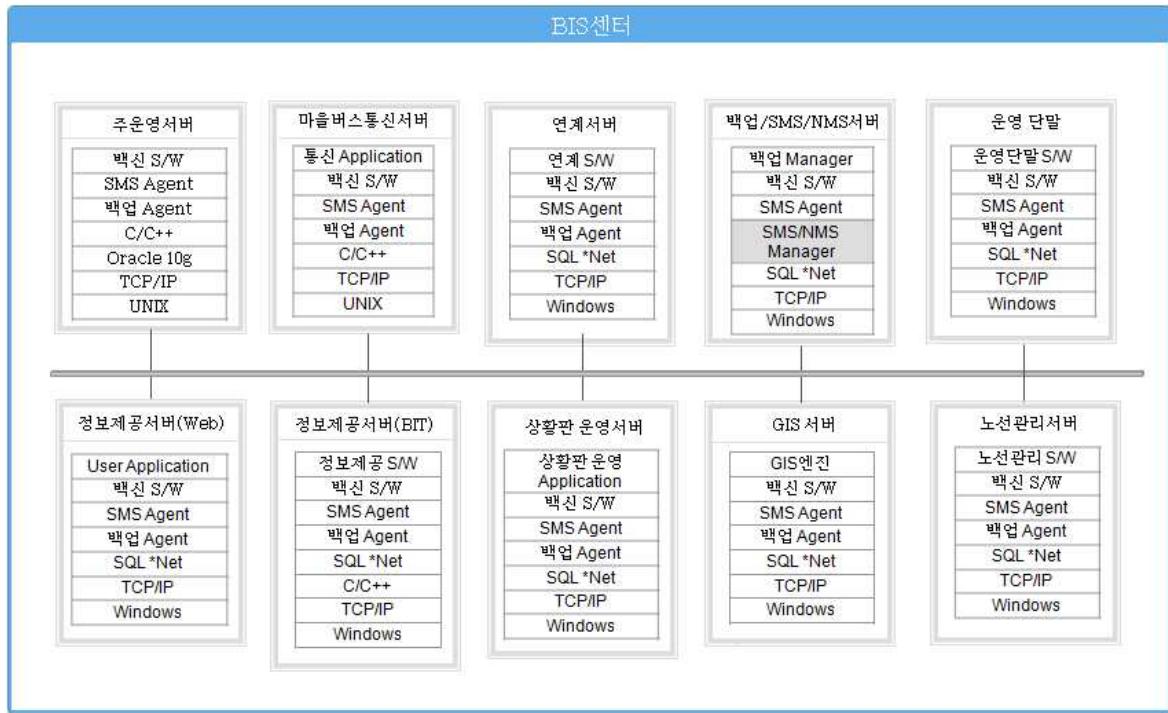
### 1) BIS 시스템 구성

- BIS 시스템이란 버스의 실시간 위치정보를 수집/가공하여 예측되어진 도착정보를 버스정류소 안내단말기 및 인터넷 정보매체를 통해 실시간으로 이용자에게 제공하는 서비스임
- 아래 그림은 현재 경기도에 구축된 시스템을 나타낸 것으로 경기도 버스종합 상황실에서 경기도 관내에 있는 모든 시내버스 및 광역버스에 대한 정보(버스 정류장 도착·출발, 교차로 통과, 돌발상황정보 등)를 수집하여 각 지자체로 전송(ASN.1)해 주면 각 지자체에서는 이를 가공하여 도착예측정보를 BIT(정류장 안내 단말기)로 전송하는 시스템으로 구성되어 있음



## 2) BIS 센터 소프트웨어

- BIS 센터 소프트웨어는 주운영서버, 통신서버, 연계서버 등을 포함하여 다음 그림과 같이 구성 됨



- 연계서버 : 경기도 BMS센터로부터 정보를 수집
- 주운영서버 : 수집된 정보를 가공하여 예측정보를 생성
- 정보제공 서버(BIT, WEB) : 이용자에게 버스 도착정보 및 생활정보를 제공
- 운영단말 : 버스관련 시설물을 관리하고 각종 운영상황을 모니터링
- 상황판 운영서버 : 시스템 및 버스 운영상황을 상황판에 표출하는 시스템
- GIS서버 : Web 및 운영단말에 지도를 관리하는 소프트웨어
- 마을버스 통신서버 : 마을버스 관련정보는 경기도 BMS센터에서 수집하지 않으므로 별도 시스템으로 정보를 수집
- 노선관리 서버 : 마을버스 관련 노선을 관리하는 프로그램
- 기타 : SMS서버, ARS서버, OPEN API, 모바일서버

### 3) ASN.1 적용 기술기준 분석

- BIS(Bus Information System)는 버스의 실시간 위치정보를 수집/가공하여 예측되어진 도착정보를 버스정류소 안내단말기 및 인터넷 정보매체를 통해 실시간으로 이용자에게 제공하는 서비스임
- BIS 구축 시 적용해야 하는 기술기준은 ‘대중교통(버스)정보교환 기술기준’으로, 기술기준에서 규정하고 있는 교환정보와 통신절차를 명확하게 숙지해야 함

<표 36> 대중교통(버스)정보교환 기술기준의 교환정보

ID	정보명	정보내용	교환주기	인터페이스
201	버스위치 정보	차량ID, 노선ID, 막차 여부, 막차의 최종 도착정류장ID • 이벤트정보 : 메시지발생시각, 수집노드ID, 노드진입·진출 시각, 노드통행시간 • 정주기정보 : 차량위치정보(GPS), 수집시각, 수집주기	실시간	센터-센터
202	도착예정 정보	정류장ID, 노선ID, 차량ID, 도착예정시간(출발정류장ID 및 진출·진입시각, 통과시간), 평균통행속도, 막차정보	실시간	센터-센터
203	운행계획 정보	노선기본정보(노선ID, 노선명칭, 기·종점정류장ID), 노선부가 정보, 노선운행정보(첫차·막차 출발시각, 첨두/비첨두 배차 간격), 차량운행회수, 운행계획정보 갱신시각·갱신내용	변경시	센터-센터
204	운행지시 정보	차량ID, 다음정차정류장ID, 차간거리조정, 운행지시정보	필요시	센터-센터
205	운행관리 정보	차량ID, 노선ID, 정보수집노드ID, 차량위치정보(GPS), 운행 상태정보, 무단결행노선ID, 무단결행발생대수, 노선명	필요시	센터-센터
206	긴급상황 정보	차량ID, 노선ID, 이벤트정보수집노드(zone, 구역)ID, 차량위치정보(GPS), 돌발상황발생시각, 돌발상황유형, 발생위치(도로명칭, 관련교차로, 돌발상황발생위치설명), 돌발상황부연 설명, 돌발상황긴급정도, 돌발상황긴급정도 부연설명	유고발생 시	센터-센터

- 또한 ‘대중교통(버스)정보교환 기술기준’은 정보교환을 위한 통신절차를 다음과 같이 규정하고 있음
  - 세션 설정
    - Client에서 로그인 패킷 전송
    - Server에서는 Sender ID, Destin ID, UserName, Password 및 각종 파라미터 적합성 여부를 판단하여 Accept를 전송하거나, Reject인 경우는 Reject 사용을 명시하여 전송

- 세션 유지

- 세션이 설정된 경우 설정된 Heartbeat duration 시간 안에 Client는 Fred 패킷을 발송하고 서버도 이에 Fred 패킷 전송

- 일정시간 안에 송수신 패킷 및 Fred 패킷이 없으면 연결 종료

- 세션 종료

- 세션은 Client 또는 Server에 의해 능동적으로 종료 가능
- Server인 경우는 종료요청(terminate-request)을, Client인 경우는 종료요청을 받거나 종료를 원할 경우 log-out 패킷을 서버로 전송

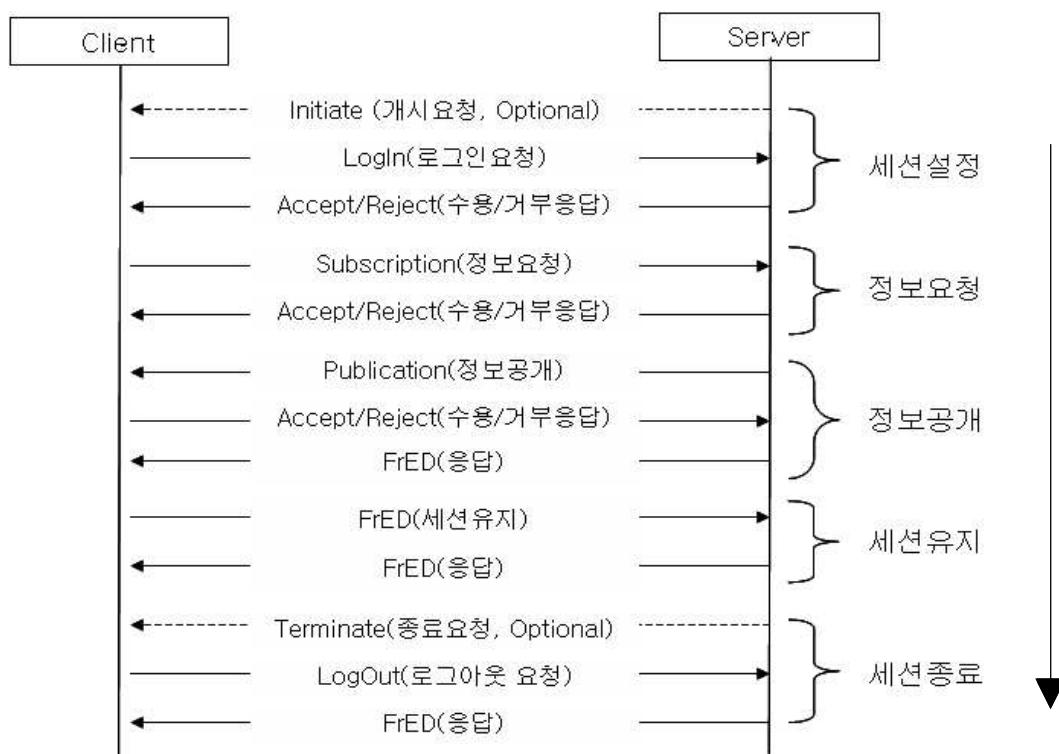
- Log-Out 패킷을 받은 서버는 Fred 패킷을 전송하고 종료

- 정보 요청

- Client에서 전송이 필요한 패킷에 대하여 한번 요청, 주기적 요청, 이벤트성 요청 인지를 명시하여 요청

- 정보 공개

- Client에서 요청한 정보가 한번 요청, 주기적 요청, 이벤트성 요청인지에 따라 데이터 전송



## 4) SAMPLE 데이터 패킷 분석

```
30 81 a2 80 01 01 81 81 98 30 81 95 80 01 03 81 01 05 82 01 01 a3 81 84 80 14 32 30
30 37 30 38 30 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
84 14 31 31 30 31 31 30 30 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 30 30 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 30 30 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a4 03 83 01 01 82 02 d5 9f
```

- T(Tag) : 0x30
- L(Length) : 0x81 0xa2
  - 데이터 길이는 0~127 범위의 값 사용
  - 데이터 길이가 128을 넘어가면 최상위 BIT를 1로 설정하고 그 외 BIT에 뒤에 올 자리수를 설정(ex : 77, 81 a2, 82 01 3d)
  - ber\_extract\_length() 함수는 Value의 길이 값만을 리턴하므로 정확한 패킷의 길이를 알고 싶을 때는 별도의 함수를 만들어 사용

	1 <sup>st</sup> byte	2 <sup>nd</sup> byte	3 <sup>rd</sup> byte	4 <sup>th</sup> byte	5 <sup>th</sup> byte	N
<b>1 byte</b>	'00' to '7F'	-	-	-	-	0 to 127
<b>2 bytes</b>	'81'	'00' to 'FF'	-	-	-	0 to 255
<b>3 bytes</b>	'82'	'0000' to 'FFFF'	-	-	-	0 to 65 535
<b>4 bytes</b>	'83'	'000000' to 'FFFFFF'	-	-	-	0 to 16 777 215
<b>5 bytes</b>	'84'	'00000000' to 'FFFFFFFF'	-	-	-	0 to 4 294 967 295

0|마지 출처 : <http://www.goferay.com/ber-tlv-length-fields/>

- V(Value)
  - 교통정보교환을 위한 국가표준(KS X ISO 14827-2)에 ASN.1으로 정의된 Datex 데이터 패킷구조 준수
    - Protocol

Header부					Data부	Tail부
Dataex Version No.	Authentication Info	Data Packet No.	Data PacketPriority No.	Header Option	PDU	CRC

DatexDataPacket ::= SEQUENCE { datex-Version-number ENUMERATED { experimental (0), version1 (1), ... } }	80 01 01 -.80 : 데이터태그(T) -.01 01 : version1 (L V)  81 81 98~~~~~02 인코딩 (T L V )
--	---

<pre>     },     datex-Data          OCTET STRING,     datex-Crc-nbr      OCTET STRING (SIZE (2)) } </pre>	82 02 d5 9f
<pre> C2CAuthenticatedMessage ::= SEQUENCE {     datex-AuthenticationInfo-text  OCTET STRING (SIZE (0..255)),     datex-DataPacket-number        INTEGER (0..4294967295),     datex-DataPacketPriority-number INTEGER (0..10),     options                      HeaderOptions,     pdu                          PDUs } </pre>	30 81 95 (T L) 80 01 03 (TLV) : Fred 81 01 05 (TLV) 82 01 01 (TLV) <b>a3</b> 81 84 ~~ (TLV) <b>a4</b> 03 83 01 01 (TLV)
<pre> HeaderOptions ::= SEQUENCE {     datex-Origin-text          UTF8String (SIZE (0..40))   OPTIONAL,     datex-OriginAddress-location OCTET STRING                OPTIONAL,     datex-Sender-text           UTF8String (SIZE (0..40))   OPTIONAL,     datex-SenderAddress-location OCTET STRING                OPTIONAL,     datex-Destination-text     UTF8String (SIZE (0..40))   OPTIONAL,     datex-DestinationAddress-location OCTET STRING          OPTIONAL,     datex-Cost                  Cost                      OPTIONAL,     datex-DataPacket-time       Time                      OPTIONAL } </pre>	
80 14 32 30 30 37 30 38 30 30 00 00 00 00 00 00 00 00 00 00 00 00 81 14 32 30 30 37 30 38 30 30 00 00 00 00 00 00 00 00 00 00 00 00 82 14 32 30 30 37 30 38 30 30 00 00 00 00 00 00 00 00 00 00 00 00 83 14 32 30 30 37 30 38 30 30 00 00 00 00 00 00 00 00 00 00 00 00 84 14 31 31 30 31 31 30 30 31 00 00 00 00 00 00 00 00 00 00 00 00 85 14 30 30 30 30 30 30 30 30 00 00 00 00 00 00 00 00 00 00 00 00	datex-Origin-text datex-OriginAddress-location datex-Sender-text datex-SenderAddress-location datex-Destination-text datex-DestinationAddress-location ※ Cost와 Time은 빠짐
<pre> PDUs ::= CHOICE {     datex-Initiate-null      [1] Initiate,     login                   [2] Login,     fred                    [3] FrED,     terminate               [4] Terminate,     logout                  [5] Logout,     subscription            [6] Subscription,     publication              [7] Publication,     transfer-done            [8] TransferDone,     accept                  [9] Accept,     reject                  [10] Reject } </pre> FrED ::= INTEGER (0..4294967295) -- datexFrED-ConfirmPacket-nbr	<b>a4</b> 03 83 01 01 (T L V) (T L V)
<pre> register unsigned short crc16 = 0; while(len--) {     crc16 = crctable[ ((crc16 ^ *buf) &amp; 0xff) ] ^ (crc16 &gt;&gt; 8);     buf++; } </pre>	ISO 3309에 정의된 CRC-16 계산

```

static unsigned short crctable[] = { // CRC-16
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241, 0xC601, 0x06C0, 0x0780, 0xC741,
    0x0500, 0xC5C1, 0xC481, 0x0440, 0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
    0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841, 0xD801, 0x18C0, 0x1980, 0xD941,
    0x1B00, 0xDBC1, 0xDA81, 0x1A40, 0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
    0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641, 0xD201, 0x12C0, 0x1380, 0xD341,
    0x1100, 0xD1C1, 0xD081, 0x1040, 0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
    0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441, 0x3C00, 0xFCC1, 0xFD81, 0x3D40,
    0xFF01, 0x3FC0, 0x3E80, 0xFE41, 0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
    0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41, 0xEE01, 0x2EC0, 0x2F80, 0xEF41,
    0x2D00, 0xEDC1, 0xEC81, 0x2C40, 0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041, 0xA001, 0x60C0, 0x6180, 0xA141,
    0x6300, 0xA3C1, 0xA281, 0x6240, 0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
    0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41, 0xAA01, 0x6AC0, 0x6B80, 0xAB41,
    0x6900, 0xA9C1, 0xA881, 0x6840, 0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBD1, 0xBC81, 0x7C40, 0xB401, 0x74C0, 0x7580, 0xB541,
    0x7700, 0xB7C1, 0xB681, 0x7640, 0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
    0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241, 0x9601, 0x56C0, 0x5780, 0x9741,
    0x5500, 0x95C1, 0x9481, 0x5440, 0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
    0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841, 0x8801, 0x48C0, 0x4980, 0x8941,
    0x4B00, 0x8BC1, 0x8A81, 0x4A40, 0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
    0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641, 0x8201, 0x42C0, 0x4380, 0x8341,
    0x4100, 0x81C1, 0x8081, 0x4040
};

```

## 5) 프로그램 샘플

```

g_DatexDataPacket = NULL;
g_AuthenticationMessage = NULL;

dc = ber_decode(NULL, &asn_DEF_DatexDataPacket, (void **) &g_DatexDataPacket, data, size);

if(dc.code == RC_OK) // Decode Success
{
    //버전정보 추출
    iVersionNbr = g_DatexDataPacket->datex_Version_number;

    memcpy(strCrc,(char *)g_DatexDataPacket->datex_Crc_nbr.buf,g_DatexDataPacket->datex_Crc_nbr.size);
    memcpy(strC2cAuth, (char *)g_DatexDataPacket->datex_Data.buf,g_DatexDataPacket->datex_Data.size);
    iC2cAuthSize = g_DatexDataPacket->datex_Data.size;

    crc16_ccitt(strC2cAuth,iC2cAuthSize, crc);
    if(crc != strCrc) return;

    dc = ber_decode(0, &asn_DEF_C2CAuthenticatedMessage,
                    (void **) &g_AuthenticationMessage,strC2cAuth,iC2cAuthSize);

    if(dc.code == RC_OK)
    {
        char     strAuthenticationInfo[256] = {0,};
        long    PacketNum, Accept_PacketNum, Reject_PacketNum, packetmask=0;
        int      Priority, psize, msk;

        // C2CAuthenticateMessage 의 datex_AuthenticationInfo_text 데이터 추출
    }
}

```

```

    memcpy(strAuthenticationInfo, (char *)g.AuthenticationMessage->datex_AuthenticationInfo_text.buf,
           g.AuthenticationMessage->datex_AuthenticationInfo_text.size);
    // C2CAuthenticateMessage 의 datex_DataPacket_number 데이터 추출
    asn_INTEGER2long(&g.AuthenticationMessage->datex_DataPacket_number, &PacketNum);
    Priority = g.AuthenticationMessage->datex_DataPacketPriority_number;

    // C2CAuthenticateMessage 의 Header 정보 추출
    if(g.AuthenticationMessage->options.datex_Origin_text->size > 0) ;
    if(g.AuthenticationMessage->options.datex-OriginAddress_location->size > 0) ;
    if(g.AuthenticationMessage->options.datex_Sender_text->size > 0) ;
    if(g.AuthenticationMessage->options.datex_SenderAddress_location->size > 0) ;
    if(g.AuthenticationMessage->options.datex_Destination_text->size > 0) ;
    if(g.AuthenticationMessage->options.datex_DestinationAddress_location > 0) ;

    switch(g.AuthenticationMessage->pdu.present)
    {
        case PDUs_PR_NOTHING: /* No components present */
        break;
        case PDUs_PR_datex_Initiate_null:
        break;
        case PDUs_PR_login: // 서버에 접속하기 위한 클라이언트의 로그인 데이터 패킷
            g_Login = g.AuthenticationMessage->pdu.choice.login;
            memcpy(sender, g_Login.datex_Sender_txt.buf, g_Login.datex_Sender_txt.size);
            memcpy(destin, g_Login.datex_Destination_txt.buf, g_Login.datex_Destination_txt.size);
            memcpy(user, g_Login.datexLogin_UserName_txt.buf, g_Login.datexLogin_UserName_txt.size);
            memcpy(passwd, g_Login.datexLogin_Password_txt.buf, g_Login.datexLogin_Password_txt.size);
            RcvHeartbeatTime = g_Login.datexLogin_HeartbeatDurationMax_qty;
            HeartbeatCount = RcvHeartbeatTime;
            RcvResponseTime = g_Login.datexLogin_ResponseTimeOut_qty;
            Datagram = g_Login.datexLogin_DatagramSize_qty;
            oid_cnt = g_Login.datexLogin_EncodingRules_id.list.count;

            break;
        case PDUs_PR_fred: // 서버와 크라라이언트의 연결을 유지하기 위한 확인 데이터 패킷
        break;
        case PDUs_PR_terminate: // 연결을 종료하고자 할때, 서버에서 클라이언트에게 요청하는 데이터 패킷
        break;
        case PDUs_PR_logout: //접속을 종료하기 위한 클라이언트의 로그아웃 데이터 패킷
        break;
        case PDUs_PR_subscription: /* 클라이언트가 서버에 정보를 요청할 경우 송신하는 데이터 패킷 */
        break;
        case PDUs_PR_publication:
        break;
        case PDUs_PR_transfer_done:
        break;
        case PDUs_PR_accept:
        break;
        case PDUs_PR_reject:
        break;
    }
}
else
{
    printf(" C2CAuthenticatedMessage DeCode Error..\n");
}
}

```

```

else
{
    printf(" DatexDataPacket DeCode Error..\n");
}

// 데이터 구조 FREE
asn_DEF_C2CAuthenticatedMessage.free_struct(&asn_DEF_C2CAuthenticatedMessage,g_AuthenticationMessage,0);
asn_DEF_DatexDataPacket.free_struct(&asn_DEF_DatexDataPacket,g_DatexDataPacket,0);

```

### 1.4.3 ASN.1 사용시 주의 할 점

#### 1) Length 산출시 주의점

- ber\_extract\_length() 함수의 리턴값은 Value의 길이만을 리턴한 값으로 Length의 가변적인 길이 값을 추가하여 산출해야 함

#### 2) Long Long형 데이터 입력

- 현재 Long Long형(4바이트 이상) 데이터를 입력할 수 있는 방법이 없으므로 기존함수를 수정하여 신규 함수를 만들어야 함

```

int AsnControl::asn_long2INTEGER_EDIT(INTEGER_t *st, long long value) {
    uint8_t *buf, *bp;
    uint8_t *p;
    uint8_t *pstart;
    uint8_t *pend1;
    int littleEndian = 1; /* Run-time detection */
    int add;

    if(!st) { return -1; }
    buf = (uint8_t *)malloc(sizeof(value));
    if(!buf) return -1;

    if(*(char *)&littleEndian) {
        pstart = (uint8_t *)&value + sizeof(value) - 1;
        pend1 = (uint8_t *)&value;
        add = -1;
    } else {
        pstart = (uint8_t *)&value;
        pend1 = pstart + sizeof(value) - 1;
        add = 1;
    }

    for(p = pstart; p != pend1; p += add) {
        switch(*p) {
            case 0x00: if((*(p+add) & 0x80) == 0)

```

```

        continue;
    break;
case 0xff: if((*(p+add) & 0x80)
    continue;
    break;
}
break;
}
for(pstart = p, bp = buf, pend1 += add; p != pend1; p += add)
    *bp++ = *p;
st->buf = buf;
st->size = bp - buf;
return 0;
}

```

### 3) UTF8 Decoding

- 한글의 경우 UTF8타입에서 입력 및 출력하는 경우 별도의 인코딩과 디코딩이 필요함

```

asd = getRouteSchedulePlanning->tpif_SubRouteNameText;
CString RoadWayName = utf8Decoding(asd->buf);
CString AsnControl::utf8Decoding(uint8_t *utf8)
{
    int size = MultiByteToWideChar(CP_UTF8, 0, (LPCSTR)(utf8), -1, NULL, 0);
    LPWSTR wStr = new WCHAR[size];

    MultiByteToWideChar(CP_UTF8, 0, (LPCSTR)(utf8), -1, wStr, size);

    CString str = W2CT(wStr);

    delete[] wStr;

    return str;
}

```

### 4) 패킷 메모리 해제

- 메모리 해제를 수행하지 않으면 기하급수적으로 메모리 사용량이 늘어나기 때문에 인코딩 또는 디코딩 후에는 항상 메모리 해제를 수행함

예)

```

asn_DEF_C2CAuthenticatedMessage.free_struct(&asn_DEF_C2CAuthenticatedMessage,g_AuthenticationMessage,0);
asn_DEF_DatexDataPacket.free_struct(&asn_DEF_DatexDataPacket,g_DatexDataPacket,0);
asn_DEF_Message0_MessageBody.free_struct(&asn_DEF_Message0_MessageBody,getMsg0,0);

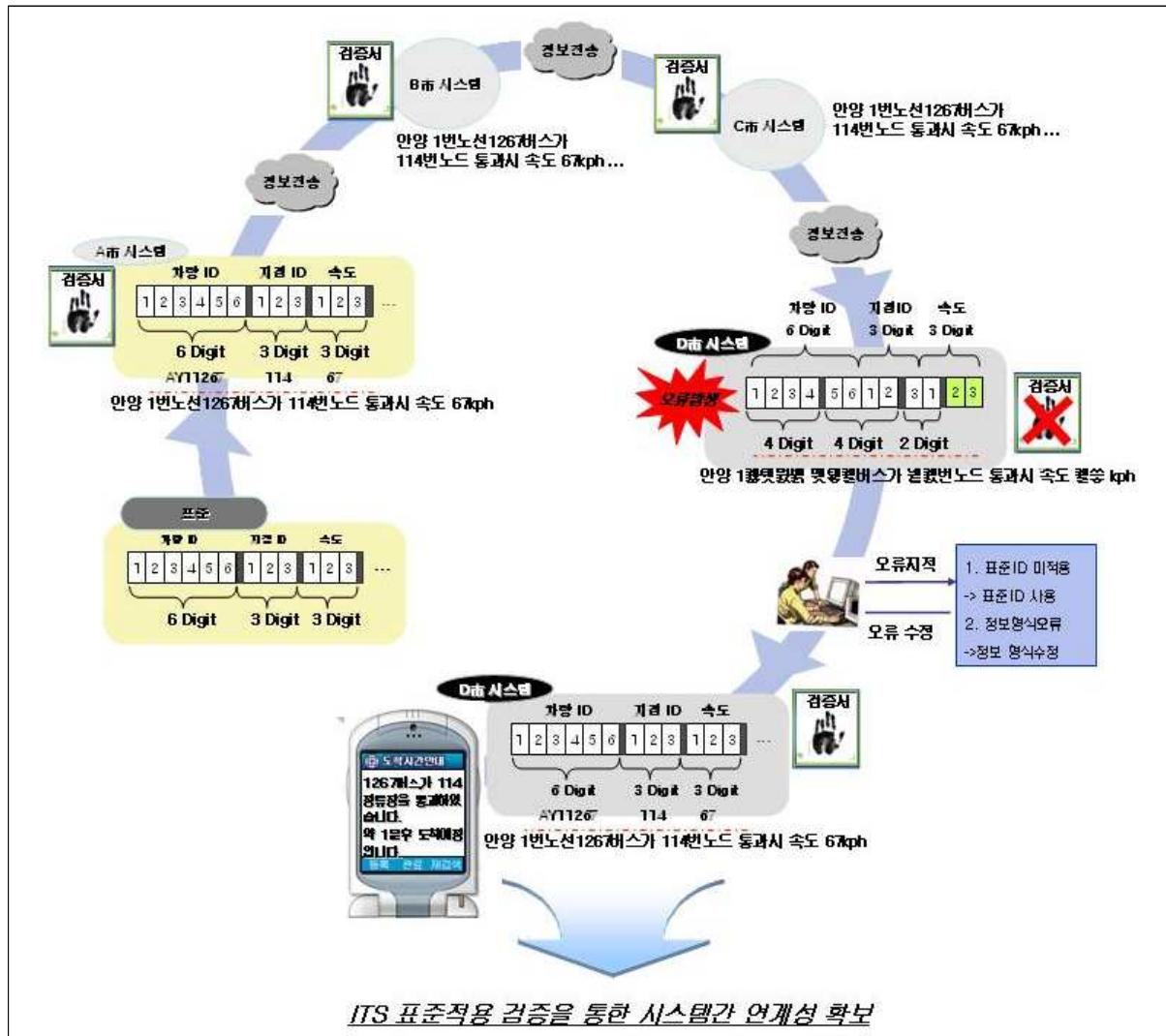
```

## 2. 표준적용 확인 방법

### 2.1 개요

#### 2.1.1 ITS 표준준수 여부 확인의 개념

- “ITS 표준준수여부 확인”은 구축 및 운영 중인 ITS 시스템의 연계성 확보를 목적으로 기술기준에 규정된 사항에 만족하는지 시험하고 판정하는 것을 의미함



<그림 36> ITS 표준준수 여부 확인의 개념

- “ITS 표준준수여부 확인”은 국가통합교통체계효율화법 제82조, 시행령 제76조 그리고 시행규칙 제31조를 법적 근거로 도로교통분야(ITS) 표준화 전담기관에서 시행하고 있음

**제82조 (지능형교통체계의 표준화)** ⑤ 교통체계지능화사업시행자는 교통체계지능화사업을 시행할 때 국토교통부장관이 관계 기관의 장과 협의하여 고시하는 바에 따라 **지능형교통체계표준을 준수하고 있는지를 확인하여야** 하며, 필요한 경우 국토교통부령으로 정하는 바에 따라 국토교통부장관에게 확인을 요청할 수 있다.

**시행령 제76조(지능형교통체계 표준의 보급)** ① 국토교통부장관은 법 제82조제5항에 따라 교통체계지능화사업시행자가 지능형교통체계표준의 준수 여부를 확인할 수 있도록 이에 관한 절차·방법·기준 등 세부 사항을 마련하여 관계 기관의 장과 협의한 후 이를 고시하여야 한다.  
② 국토교통부장관은 교통체계지능화사업시행자에 대하여 법 제82조제5항에 따라 **지능형교통체계표준의 준수 여부를 확인한 결과를 제출할 것을 요청할 수 있으며, 교통체계지능화사업시행자는 특별한 사유가 없으면 그 결과를 제출하여야 한다.**  
③ 국토교통부장관은 법 제82조제5항에 따라 교통체계지능화사업시행자로부터 요청받은 지능형교통체계표준 준수 여부의 확인 업무를 효율적으로 수행하기 위하여 제75조에 따른 지능형교통체계 표준화전담기관으로 하여금 그 업무를 대행하게 할 수 있다.

**시행규칙 제31조(지능형교통체계표준의 준수 여부 확인)** ① 법 제78조제2항에 따른 교통체계지능화사업시행자는 영 제76조제3항에 따라 지능형교통체계 표준화전담기관에 법 제82조제5항에 따른 지능형교통체계표준의 준수 여부에 대한 확인을 요청할 수 있다.  
② 영 제75조에 따른 지능형교통체계 표준화전담기관(이하 “지능형교통체계 표준화전담기관”이라 한다)은 제1항에 따른 요청을 받으면 그 사실을 지체 없이 국토교통부장관에게 통보하여야 한다.  
③ 지능형교통체계 표준화전담기관은 제1항에 따른 확인이 끝나면, 그 결과를 지체 없이 국토교통부장관과 확인을 요청한 교통체계지능화사업시행자에게 통보하여야 한다.

## 2.1.2 ITS 표준준수 여부 확인의 필요성

- 사업시행자가 ITS 표준을 적용하여 시스템을 구축하는 경우, 주관적인 표준 해석에 의해서 ITS 표준이 올바르게 적용되지 않을 수 있음
- 올바르지 않은 ITS 표준이 적용된 시스템을 운영하는 경우, 타 시스템과 연계 시 호환성에 문제가 생길 수 있기 때문에 표준을 올바르게 적용했는지 객관적으로 확인할 필요가 있음
- 이에, 국토교통부는 ITS 시스템의 경제성 및 효율성을 확보하고 전국단위의 광역화된 정보를 수집 및 제공하기 위해, 지능형교통체계표준을 제정·고시하고 사업시행자로 하여금 표준을 준수하도록 하고 있음

**제81조 (준공검사)** ① 지능형교통체계관리청이 아닌 교통체계지능화사업시행자가 교통체계지능화사업을 완료한 때에는 대통령령으로 정하는 바에 따라 지체 없이 지능형교통체계관리청에 준공보고서를 제출하여 준공검사를 받아야 한다.  
② 제1항에 따른 준공보고서를 받은 지능형교통체계관리청은 준공검사를 한 후 그 사업이 제79조에 따라 승인한 내용대로 시행되었다고 인정되는 때에는 지체 없이 준공검사확인증을 발급하여야 한다.  
③ 지능형교통체계관리청이 아닌 교통체계지능화사업시행자가 제2항에 따른 준공검사확인증을 발급받

은 때에는 제80조제1항에 따른 인·허가 등에 따른 해당 사업의 준공에 관한 검사·인가·신고 및 확인 등을 받은 것으로 본다.

④ 준공검사의 절차 및 방법, 그 밖에 필요한 사항은 대통령령으로 정한다.

**시행령 제74조(준공검사)** ① 법 제81조제1항에 따라 지능형교통체계관리청이 아닌 교통체계지능화사업 시행자가 교통체계지능화사업을 완료하였을 때에는 다음 각 호의 서류를 포함한 준공보고서를 지능형 교통체계관리청에 제출하여야 한다.

1. 준공조서(준공설계도서 및 준공사진을 포함한다)
  2. 준공 후 토지 및 시설 등의 도면
  3. 총사업비 명세서
  4. 유지·보수 계획서
  5. 법 제80조제1항에 따라 관계 행정기관의 장과 협의한 서류 및 도면
  6. 법 제82조제5항에 따른 지능형교통체계에 관한 표준의 준수 여부의 확인 결과
  7. 법 제86조제2항에 따른 지능형교통체계성능평가의 결과
  8. 그 밖에 교통체계지능화사업실시계획의 승인 내용에 대한 이행 여부의 확인에 필요한 서류
- ② 지능형교통체계관리청은 효율적인 준공검사를 위하여 필요한 경우 관계 행정기관, 「공공기관의 운영에 관한 법률」에 따른 공공기관, 연구기관, 그 밖의 전문기관 및 전문가 등에게 전문적인 조사업무를 의뢰할 수 있다.

- 표준 준수 및 이에 대한 확인은 「교통체계효율화법」이 「국가통합교통체계효율화법」으로 개정되면서 표준적용의 중요성과 강제성이 강화됨에 따라 그 필요성이 더욱 부각되었음
  - 법 개정 전에는 「ITS 업무요령」에 따라 표준적용을 권고하고 표준적용검증기관을 지정하여 표준적용을 검증하였으나, 현재는 「국가통합교통체계효율화법」에 따라 ITS 표준화전담기관에서 ITS 표준준수여부 확인을 수행하고 있음
  - 또한, 최근 통합개정된 자동차·도로교통분야 ITS 사업시행지침(국토교통부 고시 제2015-739호, 2015.10.7.)에서 표준적용을 권고하고, 관련 시스템의 표준 준수 여부 확인을 명시하고 있음
- 또한 「국가통합교통체계효율화법」은 ITS 사업 준공검사 시 제출하는 준공서류에 “ITS 표준준수여부 확인 결과”를 포함도록 하여, ITS 사업시행 시 반드시 표준준수여부 확인을 시행토록 규정하였음
- 따라서, ITS 사업을 준공하기 위해서는 반드시 ITS 표준준수여부를 확인하는 과정이 필요함

## 2.2 ITS 표준준수 여부 확인방법

### 2.2.1 ITS 표준준수 여부 확인을 위한 검사

- ITS 시스템이 표준을 준수하고 있는지 객관적·기술적으로 확인하기 위해 규정화된 방법에 따라 평가하는 행위를 “ITS 표준준수여부 확인 검사”라고 하며, 검사 결과를 토대로 표준준수 여부를 판단하는 것을 “ITS 표준준수여부 확인”이라고 함
- “ITS 표준준수여부 확인”을 수행해야 하는 대상 표준은 국토교통부에서 제정·고시한 기술기준이며, 표준준수여부 확인을 위한 검사방법은 ITS 단체표준으로 제정하여 마련하고 있음
- 이러한 “ITS 표준준수여부 확인”을 위한 검사는 사업시행자가 수행하거나 국토교통부가 지정한 ITS 표준화전담기관에 의뢰하여 실시할 수 있음

### 2.2.2 ITS 표준준수 여부 확인 대행

#### 1) 전담기관

- 국토교통부는 ITS 사업시행자가 시스템 구축·운영 시, 시스템의 호환성 및 연계성을 확보하기 위하여 「국가통합교통체계효율화법」 시행령 제75조에 의거하여 표준준수여부 확인 업무를 대행할 도로교통 분야 지능형교통체계(ITS) 표준화 전담기관으로 한국지능형교통체계협회를 지정함
  - 건설교통부 고시 제2005-139호(2005년 5월 27일)
- 또한, 「자동차·도로교통분야 ITS 사업시행 지침(국토교통부 고시 제2015-739)」 제12조와 제19조를 통해 국토교통부가 지정한 표준화 전담기관을 통해 표준준수여부 검증이 가능함을 명시하고 있음

**제12조 (시험운영)** ⑥사업시행자는 국토교통부장관이 지능형교통체계표준을 제정·고시한 경우에는 제19조에 따라 표준 준수여부를 확인하거나, 표준화전담기관에 확인을 요청하여야 하며, 확인 결과 지능형교통체계표준을 준수하고 있지 아니하는 경우에는 보완 등 필요한 조치를 하여야 한다.

**제19조 (표준적용 권고)** ⑤사업시행자는 구축된 ITS에 대하여 지능형교통체계표준 준수 여부를 확인하여야 한다. 지능형교통체계표준 준수여부 검증은 국토교통부가 지정한 표준화전담기관에 의뢰할 수 있다.

- 이에, 한국지능형교통체계협회는 검사방법에 대한 기준 및 검사툴을 개발하여 ITS 표준준수여부 확인을 위한 검사를 수행하고 있음

## 2) 검사 대상

- ITS 표준준수여부 확인 대상이 되는 시스템은 다음과 같음
  - ITS 시스템 및 서비스 구축 시 제정된 기술기준을 적용해야 하는 시스템
  - 준공 후, 확장 및 변경이 발생한 시스템
- ITS 표준준수여부 확인 검사는 ITS 사업 준공 전, ITS 사업시공 시 시험운영 및 평가를 진행하는 경우, 시스템이 변경된 경우, 시스템 진단이 필요한 경우에 실시함
  - 시스템 진단이 필요한 기준은 검사대상시스템의 변경, 기술기준의 개정 및 기타 국토교통부장관이 검사대상의 점검이 필요하다고 판단한 경우임
- 현재 한국지능형교통체계협회에서 ITS 표준준수여부 확인을 수행하고 있는 대상 기술기준은 「기본교통정보교환 기술기준(개정 2016.4.15., 국토교통부고시 제2016-206호)」 및 「대중교통(버스)정보교환 기술기준(개정 2016.4.7., 국토교통부고시 제2016-186호)」임
- 이를 위한 검사는 ITS 단체표준인 「기본교통정보교환 기술기준 적용 적합성 시험 표준(ITSK -00031)」 및 「대중교통(버스)정보교환기술기준 적용 적합성 시험 표준(ITSK-00040)」에서 규정된 방법과 절차에 따라 수행하고 있음
- 그 외에도 국토교통부에서 제정·고시한 「기본교통정보교환 기술기준Ⅱ」, 「DSRC를 이용한 ETCS 기술기준」에 대한 ITS 표준준수여부 확인을 위한 검사를 신청할 수 있음

<표 37> ITS 표준준수여부 확인 대상 기술기준 및 검사방법 표준

검사대상 기술기준	검사방법 표준(ITS 단체 표준)
기본교통정보교환 기술기준	ITSK-00031 기본교통정보교환 기술기준 적용검증 표준
대중교통(버스)정보교환 기술기준	ITSK-00040 대중교통(버스)정보교환 기술기준 적용검증 표준
기본교통정보교환 기술기준Ⅱ	ITSK-00050 기본교통정보교환 기술기준Ⅱ 적용적합성 시험표준
DSRC를 이용한 ETCS 기술기준	ITSK-00051 ETCS 기술기준 표준적합성 시험 표준

### 3) 신청 및 검사 절차

#### (가) 진행 절차

- ITS 표준준수여부 확인은 다음과 같은 절차에 따라 진행됨
- 사업시행자가 ITS 표준준수여부 확인을 ITS 표준화전담기관에 신청하면 다음과 같은 절차에 따라 진행됨
- ITS 표준화전담기관은 ITS 표준준수여부 확인이 신청되면, 시험방법 및 일정을 포함하는 시험계획서를 작성하여 해당 시스템에 대한 표준준수검사를 수행하며, 검사결과에 따라 시스템의 표준적합여부를 심사하여 결과서를 교부하게 됨
- 만약 사업시행자가 ITS 표준준수여부에 대한 확인검사를 직접 수행하는 경우, 국토교통부에서 지정한 ITS 표준화전담기관에서 수행하는 시험방법에 따라 평가하여야 하며, 해당 기술에 대한 시험방법이 없거나 확정되지 않았을 때는 사업시행자가 시험방법을 설정하여 ITS 표준화전담기관의 검토를 받아 수행하여야 함



<그림 37> ITS 표준준수여부 확인 신청 및 진행 절차

## (나) 신청 방법 및 구비서류

- ITS 표준화전담기관에 ITS 표준준수여부 확인을 위한 검사를 의뢰하고자 할 경우에는, 한국지능형교통체계협회 홈페이지([www.itskorea.kr](http://www.itskorea.kr))에 방문하여 신청 서식을 다운로드 및 작성하여 접수하면 됨
  - 신청 시에는 신청서식과 함께 아래 구비서류를 첨부하여 제출해야 함
  - 신청 접수처 : 오세완 선임연구원(이메일 : [galsan20@itskorea.kr](mailto:galsan20@itskorea.kr) 또는 팩스 : 031-502-0547)

〈표 38〉 ITS 표준준수여부 확인 검사 신청 시 첨부서류

ITS 표준준수여부 확인 검사 신청 시 첨부서류	1. 신청서 1부 2. 사업등록증 사본 1부(무료검증 시 해당 없음) 3. 시스템 구성도, 소프트웨어 구성도 각 1부
시스템 진단 신청 시 첨부서류	1. 신청서 1부 2. 시험성적서 사본 1부 3. 검증서 사본 1부(검증기관으로부터 발급받은 경우에 해당) 4. 검증평가결과서 사본 1부(사업시행자가 직접 검증을 수행한 경우에 해당) 5. 사업자등록증 사본 1부(무료검증 시 해당 없음) 6. 시스템 구성도, 소프트웨어 구성도 각 1부 7. 검증대상의 수정, 변경, 추가 내용을 기재한 설명서

## (다) 검사 비용

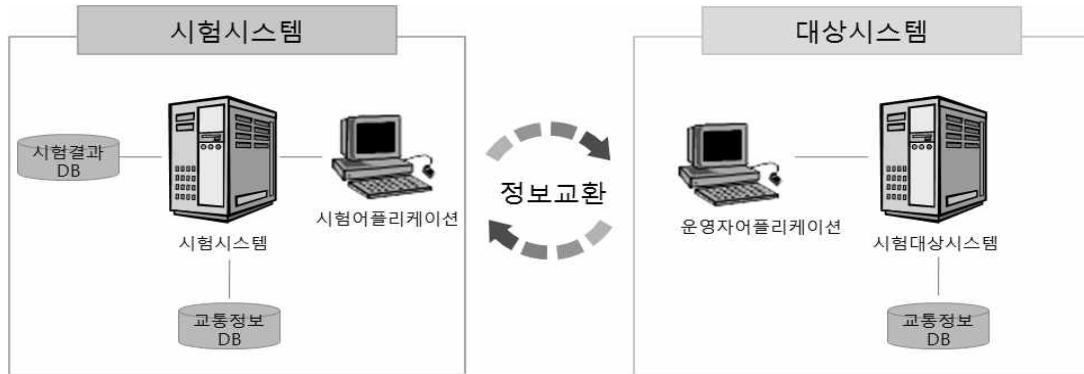
- 검사비용은 원칙적으로 유료로 진행되나 다음 항목을 충족하는 경우, 국토교통부의 지원으로 1회에 한하여 무료로 검증을 진행할 수 있음 (단, 국가의 ITS 사업이 아닌 경우와 재검사의 경우 유료로 시행)
  - 국가의 ITS 사업일 경우
  - 해당 사업의 발주기관에서 신청서를 접수할 경우
  - 해당 사업의 발주기관이 ITS 표준화전담기관인 한국지능형교통체계협회의 회원사 일 경우

## (라) 검사 방법

### ① 시험환경구성

- 표준준수여부에 대한 확인을 위해 시험대상(이하, 대상시스템)과 시험을 위해

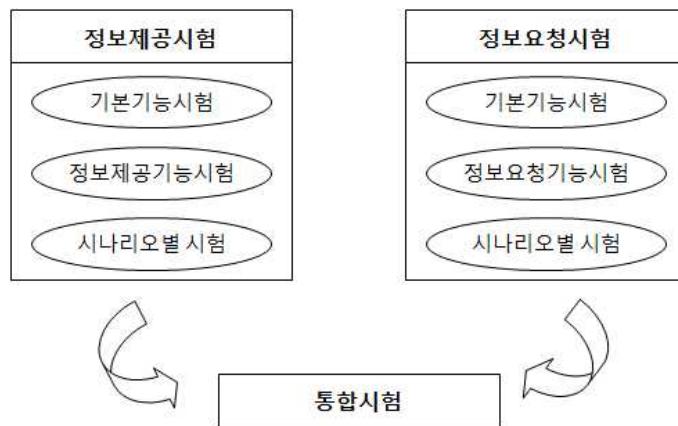
준비한 시스템(이하, 시험시스템)을 아래 그림과 같이 원격으로 시험할 수 있도록 시험환경을 구성함



<그림 38> 시험환경구성

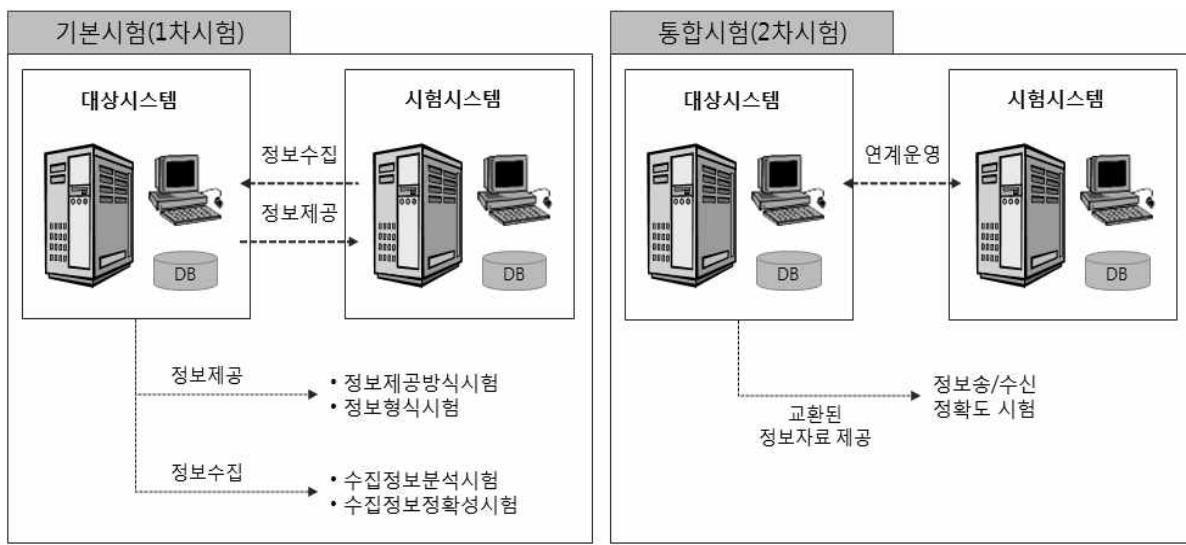
## ② 시험항목

- 시험항목은 대상시스템이 기술기준의 요구사항에 적합하게 정보를 교환하는지를 시험하기 위하여 서버기능을 시험하는 정보제공시험, 클라이언트 기능을 시험하는 정보요청시험, 시험대상의 기능을 종합적으로 시험하기 위한 통합시험으로 분류함
- 정보제공시험과 정보요청시험은 각각 정보교환을 위한 기본기능시험과 정보제공/교환 기능시험, 다양한 시나리오에 따른 시험으로 세분화됨



<그림 39> 시험항목 분류

- 통합시험은 기본기능이 제대로 동작하여야 시험 진행이 가능하므로 기본시험을 통과한 대상시스템에 한하여 실시함



<그림 40> 기본시험과 통합시험 개념도

### ③ 세부시험방법

- 정보제공시험방법은 아래와 같음
  - 정보제공시험은 대상시스템이 타 센터로부터 교통정보를 요청을 받았을 경우 기술기준의 요건에 맞게 정보의 제공이 가능한가를 시험하는 것으로써 대상시스템은 서버의 역할을 수행
  - 정보제공시험은 기본기능 시험, 정보제공기능 시험, 시나리오별 시험으로 세분화
  - 기본기능시험은 타 센터에 정보제공을 하기 위하여 세션을 설정, 유지, 종료하는 기능에 대한 시험
  - 정보제공기능시험은 대상시스템이 시험시스템으로부터 송신된 서브스크립션에 의해 요청된 9가지 기본교통정보를 기술기준에서 정의하고 있는 방법과 표준정보형식에 따라 제공하는지 확인하는 시험
  - 시나리오별 시험은 정보제공 시 발생될 수 있는 다양한 환경 하에서 대상시스템이 기술기준에서 정의하고 있는 방법 및 요구사항에 따라 이를 처리하는지 확인하는 시험

〈표 39〉 정보제공시험항목

시험항목	세부항목
정보제공을 위한 기본시험	<ul style="list-style-type: none"> <li>‣ 세션설정</li> <li>‣ 유효하지 않은 로그인정보에 대한 세션설정 거부</li> <li>‣ 비정상적 세션연결 감지</li> </ul>
교통정보 제공 기능시험	<ul style="list-style-type: none"> <li>‣ 기술기준에서 정의된 교통정보 제공기능</li> </ul>
시나리오별 시험	<ul style="list-style-type: none"> <li>‣ 중복로그인 정보 처리</li> <li>‣ 서브스크립션 분석</li> <li>‣ 서브스크립션에 의한 퍼블리케이션의 명시적 취소</li> <li>‣ 퍼블리케이션 재전송</li> </ul>

○ 정보요청시험방법은 아래와 같음

- 대상시스템이 기술기준에서 정의된 절차 및 방법으로 타 센터에 정보를 요청하여 정보를 수집할 수 있는지를 시험하는 것으로써 대상시스템은 클라이언트의 역할을 수행
- 정보요청시험은 기본기능시험, 정보요청기능시험, 시나리오별 시험으로 세분화
- 기본기능시험은 타 센터에 정보를 요청하기 위해 대상시스템이 세션 설정을 요청하고 유지하고 종료하는 기능에 대한 시험
- 정보요청기능시험은 기술기준에서 정의된 방식에 따라 타 센터에 정보를 요청하여 정보를 수집하며, 수신된 정보를 정확하게 분석할 수 있는지 확인하는 시험
- 시나리오별 시험은 정보요청 시 발생될 수 있는 다양한 환경 하에 대상시스템이 기술기준에서 정의하고 있는 방법 및 요구사항에 따라 이를 처리하는지 확인하는 시험

〈표 40〉 정보요청시험항목

시험항목	세부항목
정보요청을 위한 기본시험	<ul style="list-style-type: none"> <li>‣ 세션설정</li> <li>‣ 로그인패킷의 재전송</li> </ul>
교통정보 요청을 위한 기능시험	<ul style="list-style-type: none"> <li>‣ 기술기준에서 정의된 교통정보 요청기능</li> </ul>
시나리오별 시험	<ul style="list-style-type: none"> <li>‣ 서브스크립션 재전송</li> <li>‣ 유효하지 않은 퍼블리케이션 패킷</li> </ul>

○ 통합시험방법은 아래와 같음

- 시험시스템과 대상시스템은 두 개의 병행세션을 구성 가능
  - 시험시스템이 클라이언트, 대상시스템이 서버의 역할을 하는 세션

- 시험시스템이 서버, 대상시스템이 클라이언트 역할을 하는 다른 하나의 세션
- 통합시험은 시험시스템과 대상시스템이 병행세션을 설정하여 정보를 상호 교환하는 방식으로 진행되며, 시험시스템과 대상시스템 간에 정보제공 및 요청을 기술기준에 부합하게 처리하는지 확인
- 통합시험 항목으로는 세션유지, 정보제공통합, 정보요청통합, 다중세션 설정, 병행세션설정이 있음

#### **4) 전담기관 의뢰에 따른 장점**

- ITS 표준을 준수하면 별국가적인 ITS 시스템의 호환 및 상호연계가 가능하여 ITS 서비스의 질적 향상이 가능할 뿐만 아니라, ITS 시스템에 표준화된 기술을 적용함으로써 기술개발의 중복 투자 방지를 할 수 있음
- 이러한 기대효과가 가능하기 위해서는 ITS 표준을 올바르게 적용해야 하며, ITS 표준준수여부 확인을 통해 이를 가늠할 수 있음
- 특히, 도로교통분야 ITS 표준화전담기관인 한국지능형교통체계협회는 2005년에 ITS 표준적용검증기관으로 지정받아 표준적용 여부를 검사를 수행하여 관련 전문기술을 보유하고 있을 뿐만 아니라 약 120여건의 검사를 수행한 공신력 있는 기관으로써, ITS 사업시행자는 ITS 표준준수여부 확인을 전담기관에 의뢰하면 검사결과 및 판정에 객관성과 신뢰성을 확보할 수 있음

# 참고문헌

1. 국토교통부(2016), “국가통합교통체계효율화법(법률 제14079호)”
2. 국토교통부(2016), “국가통합교통체계효율화법 시행령(대통령령 제27205호)”
3. 국토교통부(2016), “국가통합교통체계효율화법 시행규칙(국토교통부령 제357호)”
4. 국토교통부(2016), “기본교통정보교환 기술기준”
5. 국토교통부(2016), “기본교통정보교환 기술기준(Ⅱ)”
6. 국토교통부(2016), “기본교통정보교환 기술기준(IV)–무선통신 기술을 이용한 교통정보 수집·제공 기술표준”
7. 국토교통부(2016), “대중교통(버스) 정보교환 기술기준”
8. 국토교통부(2013), “근거리 전용통신(DSRC)를 이용한 자동요금징수시스템(ETCS)의 정보교환 기술기준(노면–단말간)”
9. 국토해양부(2010), “자동차·도로교통분야 국가 ITS 아키텍처(ver2.0)”, 국토해양부
10. 건설교통부(2008), “기본교통정보교환 기술기준 적용방법 해설서 Ⅱ”, 건설교통부
11. 건설교통부(2007), “기본교통정보교환 기술기준 적용방법 해설서”, 건설교통부
12. 건설교통부(2007), “기본교통정보교환 기술기준” – 교육자료, 건설교통부
13. ITS 국가교통정보센터 웹사이트 (<http://www.its.go.kr/>)

# **기술기준의 이해 및 적용을 위한 심화교육**

---

**초 판 | 2005년 11월 15일 발행**

**개정판 | 2016년 10월 19일 발행**

**편 저 사단법인 한국지능형교통체계협회**

경기도 안산시 상록구 성호로 31 ITS 인증·성능평가센터

Tel. 031-478-0447 Fax. 031-502-0547

Homepage. <http://www.itskorea.kr>

**발행처 국토교통부**

세종특별자치시 도움6로 11 국토교통부

**인쇄처 (주)새론기획**

---

※ 무단 복사·복제를 금합니다.