

(사업명: 클라우드 엣지 기반 도시교통 브레인 핵심기술 개발)

# 교통신호 최적화 서브시스템 설계서

## V2.0

스마트데이터연구실

지능정보연구본부 초지능창의연구소

 한국전자통신연구원  
[www.etri.re.kr](http://www.etri.re.kr)

## 문서 정보

구분	소속	성명	비고
작성자			
승인자			
버전	2.0		
승인(발행)일	2023.11.13.		
상태	동료검토완료		

\* 상태: 초안, 동료검토완료, 승인완료

## 개정 이력

버전	개정일자	개정내역	작성자	승인자
1.0	2021.11.30.	초안작성		
1.5	2022.12.9.	인터페이스 보완, 불필요 기능 삭제		
2.0	2023.11.13.	인터페이스 수정 반영, 전이학습 관련 보완, 교통 환경 연동 기능 삭제		

## 목차

1		
2	<b>1. 개요</b>	<b>7</b>
3	1.1 목적	7
4	1.2 범위 및 구성	7
5	1.3 용어 및 약어	7
6	1.3.1 용어 정의	7
7	1.3.2 약어	8
8	1.4 참고 문헌	8
9	<b>2. 서브 시스템 설계 고려사항 및 목표</b>	<b>9</b>
10	2.1 설계 고려사항	9
11	2.2 설계 목표	10
12	<b>3. 서브 시스템 구조</b>	<b>11</b>
13	3.1 강화학습 기반 신호 최적화 블록	11
14	3.2 최적 설계 기반 신호 최적화 블록	12
15	<b>4. 서브 시스템 기능 설계</b>	<b>14</b>
16	4.1 교통 신호 최적화를 위한 모사 교통 환경 활용 강화학습	14
17	4.1.1 교통 신호 최적화 정책(강화학습 모델)	15
18	4.1.2 강화학습시 상태 설계	16
19	4.1.3 강화학습시 보상 설계	17
20	4.1.4 강화학습 시 행동 설계	18
21	4.1.5 교통 신호 최적화 강화학습을 위한 도커 이미지 제공	23
22	4.2 대규모 교통 네트워크 신호 최적화 분산 처리	24
23	4.2.1 교차로 그룹핑	24
24	4.2.2 다중 에이전트 기반 신호 최적화	24
25	4.3 전이학습 기반 신호 최적화 강화학습 모델 생성	26
26	4.3.1 사전 학습 훈련 및 사전 학습 모델 생성	27
27	4.3.2 전이 학습 훈련 및 전이 학습 모델 생성	27
28	4.4 최적 설계기반 교통 신호 최적화	28
29	4.4.1 최적 신호 설계를 위한 교통 데이터 분석	28
30	4.4.2 단일 교차로에 대한 신호 현시 분할	28

1	4.4.3 다중 교차로에 대한 연동축 그룹핑 .....	28
2	4.4.4 연동 교차로 신호간 신호 오프셋 산출 .....	29
3	<b>5. 서브 시스템 인터페이스.....</b>	<b>30</b>
4	5.1 교통 신호 최적화 실행 인터페이스.....	30
5	5.1.1 교통 신호 최적화 실행 프로그램 직접 실행 .....	30
6	5.1.2 클라우드 서비스를 위한 도커 이미지 활용 .....	34
7	5.2 신호 현시 분할 제공 인터페이스 .....	34
8		
9		
10		

## 그림 목차

그림 1	교통신호 최적화 서브시스템 구성도 .....	11
그림 2	교통 신호 최적화를 위한 모사 교통 환경을 활용한 강화학습 개념 .....	15
그림 3	독립형 멀티 에이전트 활용 개념 .....	15
그림 4	협업형 멀티 에이전트 활용 개념 .....	16
그림 5	보상 수집 .....	18
그림 6	모델 출력의 신호 페이즈 녹색시간으로 가공 예 .....	19
그림 7	사이클 길이 제약을 만족시키는 모델 출력의 가공 예 .....	20
그림 8	모델 출력의 신호 사이클의 녹색시간으로 가공 예 .....	20
그림 9	지자체 신호 조정 방법에 기반한 현실 제약 조건을 모두 만족시키는 신호 조정 예 .....	22
그림 10	현시 조정 테이블을 활용한 모델 출력의 신호사이클 녹색시간으로 가공 예 .....	23
그림 11	다중 에이전트기반 신호 최적화 개념 .....	25
그림 12	교통 신호 최적화 에이전트 할당 개념 .....	26
그림 13	전이학습 기반 신호 최적화 개념 .....	27
그림 14	신호 현시 분할 과정 .....	28
그림 15	연동축 구간 선정 및 연동 오프셋 산출 개념도 .....	29
그림 16	신호 TOD 분할 결과 예 .....	34

## 1. 개요

### 1.1 목적

본 문서는 “클라우드 엣지 기반 도시교통 브레인 핵심기술 개발” 최종 결과물인 도시교통브레인(코드명은 UNIQ: Urban Network Intelligence for solving traffic Queues, 이하 UNIQ)을 구성하는 교통신호 최적화 서브시스템의 설계 내용을 정리한 문서이며, 향후 교통신호 최적화 서브시스템의 설계 내용과 인터페이스에 대한 이해를 돕기 위한 문서로 활용하고자 한다.

### 1.2 범위 및 구성

본 서브시스템 설계서는 대도시의 교통소통 최적화를 목적으로 클라우드-엣지 기반 실시간 교통상황 분석 및 대규모 교통 시뮬레이션을 통해 교통제어 지능을 제공하는 도시교통 브레인을 개발하기 위해 수집한 요구사항들을 정리한 UNIQ 요구사항 정의서[3] 를 기반으로 시스템 개략 설계 내용을 정리한 도시교통브레인 시스템 설계서[4]에서 도출된 서브시스템들 중 교통신호 최적화 서브시스템의 상세 설계 내용을 정리한 문서이다.

본 문서의 구성은 다음과 같다. 2장에서는 서브시스템 설계 고려사항 및 목표에 대해 기술하고, 3장에서는 전체 서브시스템 구조와 서브시스템을 구성하는 각 블록의 역할에 대한 정의 및 각 블록 간의 관련성과 인터페이스에 대해 기술한다. 4장에서는 3장에서 분류된 각 블록이 제공하는 기능과 이를 제공하기 위한 설계에 대하여 기술하고, 5장에서는 교통신호 최적화 서브시스템이 제공하는 인터페이스에 대하여 기술한다.

### 1.3 용어 및 약어

#### 1.3.1 용어 정의

- 강화학습: 현재의 상태(State)에서 어떤 행동(Action)을 취하는 것이 최적인지를 학습하는 것으로, 행동을 취할 때마다 외부 환경에서 보상(Reward)이 주어지는데, 이러한 보상을 최대화하는 방향으로 학습이 진행됨
- 교통공학: 교통의 기능과 안전성의 향상, 주변 환경과의 조화 등을 공업적인 부분에서 연구하는 학문으로, 각종 교통시설물의 계획, 설계, 운영, 관리에 필요한 기술과 과학적 원칙을 적용하는 것을 포함함.

- 교통혼잡: 도로용량을 초과하는 과다한 교통 수요나 도로구조상의 문제, 그리고 교통사고 등의 원인에 의해서 발생하는 차량의 지·정체 및 대기행렬 현상
- 도시교통 브레인: 교통 데이터 분석을 통하여 도시 교통흐름 최적화를 수행하는 인공지능 핵심기술
- 도커: 리눅스 컨테이너 생성 및 사용을 돕는 컨테이너 기술
- 신호 분할: 하루를 유사한 통행 특성을 갖는 시간 그룹으로 분할하는 것
- 클라우드-엣지: 중앙 데이터센터 중심의 기존 클라우드 시스템에 엣지컴퓨팅 기술을 적용·보완하여 고도화한 데이터 활용을 위한 플랫폼

### 1.3.2 약어

API	Application Programming Interface
JSON	Java Script Object Notation
SALT	Simulation for AnaLyzing urban Traffic
SUMO	Simulation of Urban MObility
SW	Software
TOD	Time Of Day

## 1.4 참고 문헌

- [1] ETRI, 클라우드 엣지 기반 도시교통 브레인 핵심기술 개발 사업수행계획서(v4.0), 2023
- [2] ETRI, 한국전자통신연구원 연구개발 표준 프로세스(v4.5), 2023
- [3] ETRI, 도시교통 브레인 요구사항정의서(v2.0), 2023
- [4] ETRI, 도시교통 브레인 시스템 설계서(v3.0), 2023
- [5] 도커 명령어, <https://docs.docker.com/engine/reference/commandline/docker/>
- [6] KAIST, 최적 설계기반 교통 신호 최적화 블록 설계서(v1.0), 2021



## 2. 서브 시스템 설계 고려사항 및 목표

본 서브시스템 설계의 범위는 당해년도에 수행되는 연구 개발로 그 범위를 한정하는 것이 아니라 전체 연구 기간에 수행될 연구 개발 내용을 모두 포함한다.

### 2.1 설계 고려사항

교통신호 최적화 서브시스템 설계시 다음과 같은 사항에 대해 고려 한다.

- 민간/공공 교통 데이터 활용  
민간/공공에서 생성된 다양한 이종의 교통 데이터를 대도시의 교통소통 최적화에 활용한다.
- 개인 정보 보호  
민간 교통데이터, 공공 교통데이터 활용시 개인 정보 침해와 관련한 문제의 소지가 없는지 미리 검토한다.
- 가상 교통 환경 활용  
교통 시뮬레이터를 활용하여 가상의 교통 환경을 생성하고 이를 활용한 다양한 실험을 수행하여 충분히 검증한 후에 실세계의 교통 환경에 적용한다.
- 교통 환경의 현실 제약 고려  
최소 녹색시간 보장, 현시 순서 보장, 신호 주기 보장 등과 같은 교통 환경의 현실 제약 조건을 고려하여 교통신호를 최적화한다.
- 공개 소프트웨어 활용  
가용한 공개 소프트웨어를 최대한 활용하고, 시스템의 호환성을 유지한다. 활용할 공개 SW 선정시 Apache 2.0 라이선스와 호환성이 있는지 확인한다.
- 파이썬 기반 개발 고려  
교통 신호 최적화 서브시스템은 파이썬 기반으로 개발함을 고려한다.

## 2.2 설계 목표

교통신호 최적화 서브시스템 설계시 2.1절에서 언급한 내용을 고려하고 아래와 같은 설계 목표들이 달성될 수 있도록 진행한다.

- 대도시 규모의 교통 네트워크 환경에서도 신호 최적화에 기반한 교통 혼잡 제어 지원
- 교통 신호 최적화 과정에서 시민들의 불편이 최소화될수 있도록 가상 교통 환경에 기반한 강화학습을 통해 교통 신호 최적화 지원
- 다양한 교통 환경에서 교통신호 최적화 지원
- 클라우드 서비스를 활용한 도심 교통 혼잡 제어 지원
- 지자체 신호제어 및 교통 인프라에 적용을 통한 교통신호 최적화 실증
- 성능 목표
  - 교차로 통과시간 감소율 15% 이상
  - 200개 이상의 교차로에 대한 신호 최적화 가능

### 3. 서브 시스템 구조

교통신호 최적화 서브시스템은 수집한 교통 데이터와 교통 시뮬레이터를 활용하여 교통 공학 이론뿐 아니라 강화학습을 적용하여 교통 신호 최적화를 수행하는 서브 시스템이다. 본 장에서는 교통 신호 최적화 서브시스템의 상세 구조에 대해 설명한다.

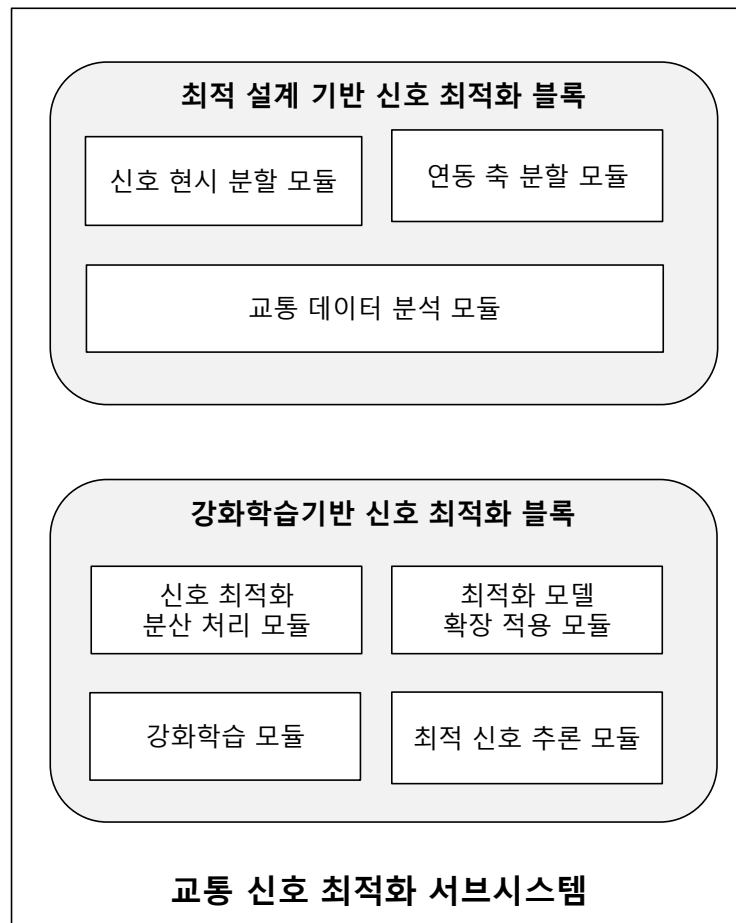


그림 1 교통신호 최적화 서브시스템 구성도

교통신호 최적화 서브시스템의 세부 구성은 그림 1에서 보는 바와 같이 강화학습 기반 신호 최적화 블록, 최적 설계 기반 신호 최적화 블록으로 나누어 진다.

#### 3.1 강화학습 기반 신호 최적화 블록

강화학습 기반 신호최적화 블록은 실제 도심 교통 환경을 모사한 교통 시뮬레이터를 활용하여 강화학습 기반으로 신호 최적화를 수행하는 블록으로, 다음과 같은 기능을 제공한다.

- 교통 신호 최적화를 위한 모사 교통 환경 활용 강화학습

- 대규모 교통 네트워크 신호 최적화 분산 처리
- 사전학습 모델 활용 신호 최적화 강화학습 모델 생성
- 교통 혼잡 해소를 위한 강화학습 모델 기반 최적 신호 추론

이를 위해 강화학습기반 신호 최적화 블록은 아래와 같이 강화학습 모듈, 신호 최적화 분산 처리 모듈, 최적화 모델 확장 적용 모듈, 최적 신호 추론 모듈의 4개의 모듈로 구성된다.

- 강화학습 모듈  
도시 교통 시뮬레이션 환경에 기반한 강화학습을 통해 도시 교통을 최적화하기 위해 머신러닝 모델의 훈련 및 훈련된 모델의 관리 등을 담당하는 모듈이다.
- 신호최적화 분산처리 모듈  
다수의 에이전트와 다수의 클라우드 엣지에 기반한 분산 처리를 통해 대규모 교통 네트워크에 대한 신호 최적화가 가능하게 하기 위한 모듈이다.
- 최적화 모델 확장 적용 모듈  
교통신호 최적화를 위해 이미 훈련된 머신러닝 모델을 활용하여 교통 특성이 유사한 교차로 그룹에 대해 해당 교차로의 신호 최적화를 수행할수 있는 모델을 생성하는 모듈이다.
- 최적 신호 추론 모듈  
학습된 강화학습 모델에 교통 환경으로부터 수집한 실시간 교통 데이터를 입력으로 하여 단일 교차로, 다중 교차로, 교통 네트워크에 대해 최적의 교통 신호를 추론하는 모듈이다.

### 3.2 최적 설계 기반 신호 최적화 블록

최적 설계기반 신호 최적화 블록은 수집한 교통 데이터에 기반한 교통류 특성을 분석하여 교통 공학을 활용한 시간대 별 교통 흐름 변화에 따른 신호 분할, 교차로 간 신호 연동 등을 통해 교통 공학 이론에 기반한 최적 신호를 생성할 수 있게 하는 블록으로, 다음과 같은 기능을 제공한다.

- 최적 신호 설계를 위한 교통 데이터 분석
- 단일 교차로에 대한 신호 현시 분할
- 다중 교차로에 대한 연동축 그룹핑
- 연동 교차로 신호간 신호 오프셋 산출

이를 위해 최적 설계기반 신호 최적화 블록은 아래와 같이 교통 데이터 분석 모듈, 신호 현시 분할 모듈, 연동 축 분할 모듈의 3개 모듈로 구성된다.

- 교통 데이터 분석 모듈

교통 공학 이론에 기반한 신호 최적화를 위해 수집된 교통 데이터에 대해 진행 방향(직진, 좌회전, 우회전, 유턴 등)별 비율 분석, 시간대별 비율 분석, 유입 및 유출 경로 분석 등의 다양한 분석을 제공하는 모듈이다.

- 신호 현시 분할 모듈

수집한 교통 데이터에 대한 진행 방향별 비율, 시간대별 비율 등의 분석 결과를 전체 통행량 형태로 가공한 후 유사한 통행 특성을 같은 시간 그룹으로 군집하여 단일 교차로에 대한 신호 현시를 분할하는 모듈이다.

- 연동 축 분할 모듈

주요 교차로에 대한 유입/유출 경로 분석 결과에 기반하여 연속된 주요 구간을 하나의 연동축으로 분할 후 신호 그룹으로 그룹핑하여, 차량이 해당 구간을 신호 지체없이 통과할 수 있도록 신호 그룹 내 다중 교차로의 신호에 대해 적정 오프셋(offset)을 산정하는 모듈이다.

## 4. 서브 시스템 기능 설계

본 장에서는 교통 신호 최적화 서브시스템이 제공하는 주요 기능과 이를 제공하기 위한 설계에 대해 다음과 같이 분류하여 설명한다.

- 교통 신호 최적화를 위한 모사 교통 환경 활용 강화학습
  - 교통 신호 최적화 정책(강화학습 모델)
  - 강화학습시 상태
  - 강화학습시 보상
  - 강화학습시 행동
  - 교통신호 최적화 강화학습 도커 이미지
- 대규모 교통 네트워크 신호 최적화 분산 처리
  - 교차로 그룹핑
  - 다중 에이전트 기반 신호 최적화
- 기학습 모델 활용 신호 최적화 강화학습 모델 생성
  - 교통 특성 유사 교차로 그룹핑
  - 모델 전이를 통한 교통 신호 최적화를 위한 강화학습 모델 생성
- 최적 설계 기반 신호 최적화
  - 최적 신호 설계를 위한 교통 데이터 분석
  - 단일 교차로에 대한 신호 현시 분할
  - 다중 교차로에 대한 연동축 그룹핑
  - 연동 교차로 신호간 신호 오프셋 산출

### 4.1 교통 신호 최적화를 위한 모사 교통 환경 활용 강화학습

교통 신호 최적화를 위해 도시 교통 환경을 모사한 시뮬레이션을 활용한 강화학습을 수행한다. 그림 2는 교통 신호 최적화를 위해 모사 교통 환경(교통 시뮬레이터)을 활용한 강화학습의 개념도이다.

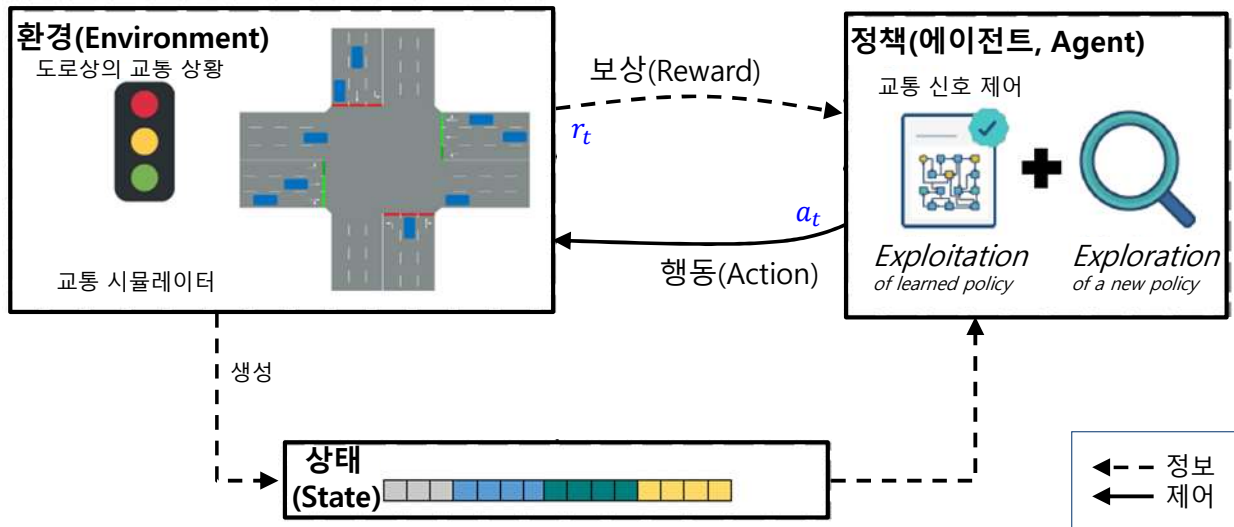


그림 2 교통 신호 최적화를 위한 모사 교통 환경을 활용한 강화학습 개념

교통 신호 최적화를 위한 모사 교통 환경 활용 강화학습을 위한 상세 기능의 설계는 다음과 같다.

#### 4.1.1 교통 신호 최적화 정책(강화학습 모델)

가) 독립형 에이전트를 활용한 교통 신호 최적화 정책 제공

독립형 에이전트를 활용한 교통 신호 최적화를 위해서 각 교차로마다 하나의 독립된 에이전트를 두고 해당 교차로에 대한 교통 신호 최적화를 담당하게 한다. 에이전트를 구성하는 강화학습 모델로는 단일 에이전트에서 좋은 성능을 보이는 DDQN(Double Deep Q-Networks) 모델, 연속 행동 공간(Continuous Action Space)에서 좋은 성능을 보이는 DDPG(Deep Deterministic Policy Gradients) 모델, 연속 행동 공간뿐만 아니라 이산 행동 공간(Discrete Action Space)에서도 모두 좋은 성능을 보이는 PPO(Proximal Policy Optimization) 모델 등을 활용한다.

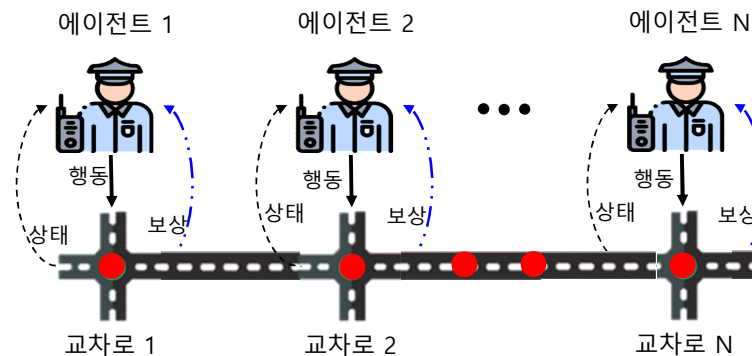
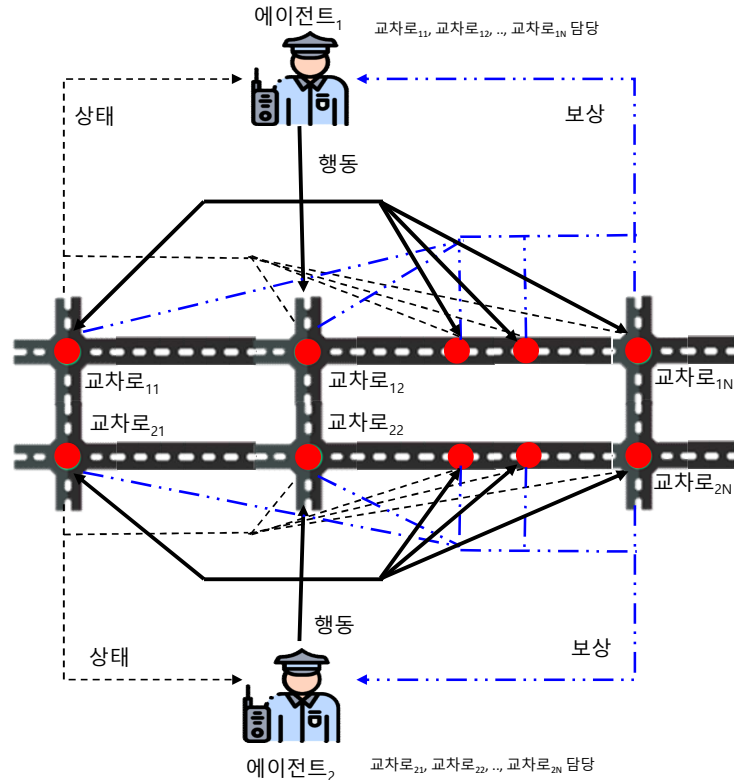


그림 3 독립형 멀티 에이전트 활용 개념

1

## 2 나) 협업형 에이전트를 활용한 교통 신호 최적화 정책 제공



3

4 그림 4 협업형 멀티 에이전트 활용 개념

5 협업형 에이전트를 활용한 교통 신호 최적화를 위해 하나의 에이전트가 인접한 다수의  
 6 교차로들에 대한 교통 신호 최적화를 담당하게 한다. 에이전트는 담당하고 있는 교차로들의  
 7 상태 정보를 입력으로 활용하여 추론을 수행한 후 각각 교차로의 행동으로 가공한다. 또한,  
 8 에이전트는 담당하고 있는 교차로들에 대해 행한 행동의 영향에 기반하여 담당하는  
 9 교차로들로부터 보상을 수집하여 모델 갱신에 활용한다. 이와 같이 하나의 에이전트가 특정  
 10 교차로에 대한 신호 최적화를 함에 있어 해당 교차로의 교통 상황뿐 아니라 주변 교차로의  
 11 교통 상황과 주변 교차로에의 영향을 고려함으로써 교통 소통을 원활하는데 있어 교차로간  
 12 협업이 될수 있도록 한다.

13 에이전트를 구성하는 강화학습 모델로는 협업형 멀티에이전트 게임에서 좋은 성능을 보인  
 14 PPO 모델을 활용한다.

15

## 16 4.1.2 강화학습시 상태 설계

17 교통 신호 최적화를 위한 강화학습 모델의 입력이 되는 상태(State)로 다음의 값의 조합을



이용할 수 있으며, 인접 교차로와 연동을 고려하기 위해 인접 교차로 정보를 포함하여 수집하기도 한다.

- 진입 차선별 밀도
- 평균 차량 속도
- 신호 교차로의 대기열 길이
- 대기 시간
- 차량 통과 수
- 신호 페이즈<sup>1</sup>

#### 4.1.3 강화학습시 보상 설계

##### 가) 보상 항목

교통 신호 최적화를 위한 강화학습 시 보상(Reward) 정보로 다음의 값들을 이용할 수 있으며, 인접 교차로와 연동을 고려하기 위해 인접 교차로 정보를 포함하여 수집하기도 한다.

- 신호 교차로의 (축적된) 대기열 길이
- 대기 시간
- 차량 통과 수
- 교차로 진입 차량 수와 진출 차량 수의 차이(pressure)
- 기존 신호 주기와 각 현시의 주기 합의 차이

##### 나) 보상 수집 방법

보상 정보는 행동을 수행 완료 후 다음 상태 정보를 수집하는 시점에 한번 수집하기도 하지만 행동 수행 중 후 다음 상태 정보 수집하기까지 일정 주기를 가지고 수집하기도 한다.

<sup>1</sup> 신호 페이즈(signal phase) : 신호 현시; 통행할 수 있는 교통류를 의미; 신호 테이블에서 현재 녹색인 것을 가리킴

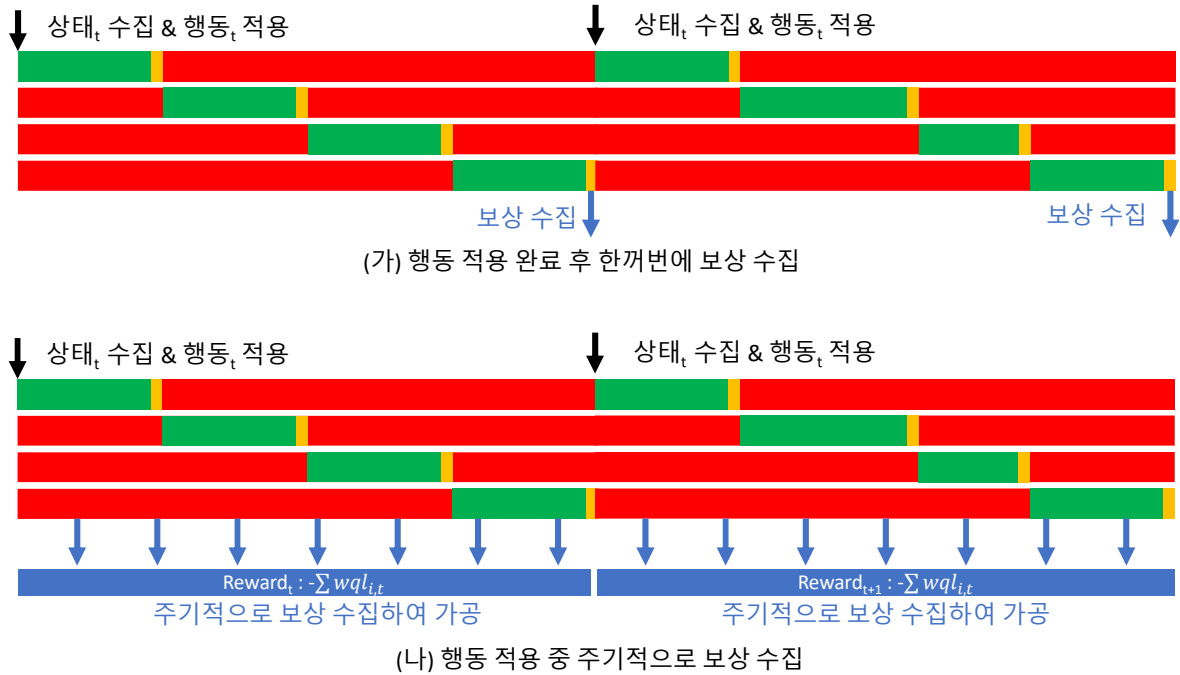


그림 5 보상 수집

#### 4.1.4 강화학습 시 행동 설계

실세계 교통 환경에서는 현시 순서, 신호 주기, 최소/최대 녹색시간과 같은 신호 시스템 설계시 고려해야 하는 현실 제약 조건이 있다. 본 항에서 교통 신호 최적화를 위한 강화학습에서 행동(Action)은 현실 제약 조건을 고려하는지 여부에 따라 아래와 같이 2 가지 유형으로 나누어서 설계를 한다.

가) 현실 제약 조건을 고려하지 않은 행동

(1) 신호 페이즈 유지 혹은 변경

강화학습 모델의 출력을 현재 신호 페이즈(Signal Phase)를 일정시간 동안 연장하여 유지하거나 신호 사이클 상의 다음 신호로 변경할지 여부를 나타내는 값으로 가공한다. 이 값에 기반하여 신호 페이즈를 유지하거나 신호 사이클 상의 다음 신호로 변경한다.

(2) 다음 녹색 신호가 부여될 신호 페이즈 선정

강화학습 모델의 출력을 신호 사이클을 구성하는 신호 페이즈들 중에 다음 일정시간 동안 부여될 신호 페이즈를 나타내는 값으로 가공한다. 가공된 신호 페이즈가 수행되도록 신호를 설정한다.

## 나) 신호의 현실 제약 조건을 고려한 행동

## (1) 최소 녹색시간을 만족시키는 다음 신호 페이즈의 녹색 시간 설정

강화학습 모델의 출력을 최소 녹색시간에 더해 할당할 추가 녹색시간으로 가공한다. 신호 페이즈의 녹색 시간을 최소 녹색시간과 추가 녹색시간의 합으로 계산한다. 이때 최대 녹색시간 만족 여부는 체크하지 않는다. 이에 기반하여 다음 신호 페이즈의 녹색시간을 설정하고 다음 신호 페이즈로 변경한다.

## (2) 최소/최대 녹색 범위 내에서 다음 신호 페이즈의 녹색시간 설정

강화학습 모델의 출력을 다음 신호 페이즈의 최소 녹색시간과 최대 녹색시간 사이의 하나의 값으로 가공한다. 이를 다음 신호 페이즈의 녹색 시간으로 설정하고 다음 신호 페이즈로 변경한다. 그림 6은 모델 출력을 다음 신호 페이즈의 녹색시간으로 가공하는 예이다.

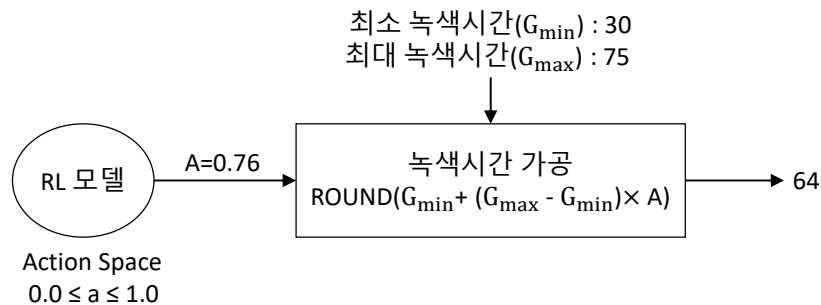


그림 6 모델 출력의 신호 페이즈 녹색시간으로 가공 예

## (3) 최소 녹색시간과 신호 사이클 길이 제약을 만족시키는 신호 페이즈의 녹색 시간 설정

강화학습 모델의 출력을 최소 녹색시간에 더해 할당할 추가 녹색시간으로 가공한다. 이를 위해 신호 사이클 길이에서 각 신호 페이즈의 최소 녹색시간의 합을 제한 후에 남는 시간을 각각의 신호 페이즈에 배분한다. 신호 페이즈의 녹색시간은 최소 녹색시간에 배분된 추가 녹색시간을 더한 시간이 된다. 이때 최대 녹색시간 만족 여부는 체크하지 않는다. 이에 기반하여 다음 신호 사이클 동안 수행될 신호를 설정한다. 그림 7는 사이클 길이 제약을 만족시키는 녹색시간 가공의 예이다.

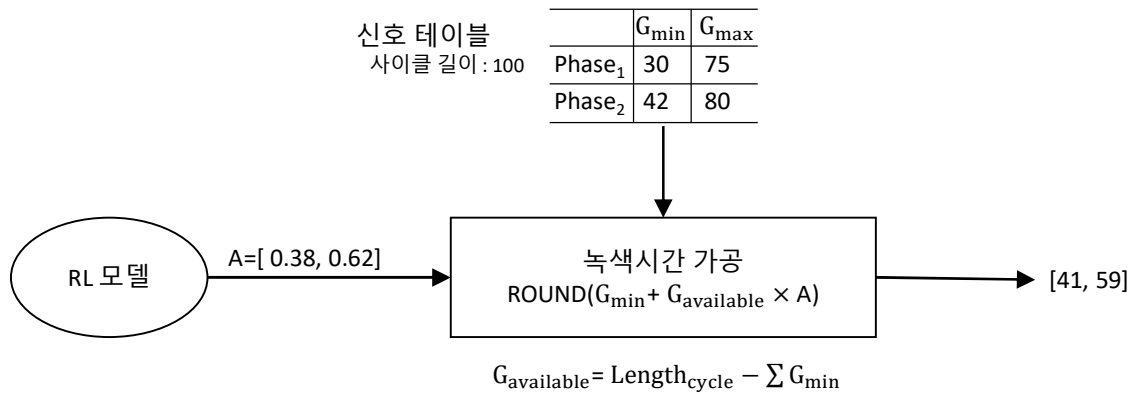


그림 7 사이클 길이 제약을 만족시키는 모델 출력의 가공 예

- (4) 최소/최대 녹색시간 범위 내에서 신호 사이클을 구성하는 신호 페이즈들의 녹색시간 설정

강화학습 모델의 출력을 각 신호 페이즈의 최소 녹색시간과 최대 녹색시간 사이의 하나의 값들로 가공한다. 이때 신호 사이클 길이 만족 여부는 체크하지 않는다. 이에 기반하여 다음 신호 사이클 동안 수행될 신호를 설정한다. 그림 8은 모델 출력을 신호 사이클의 녹색시간으로 가공하는 예이다.

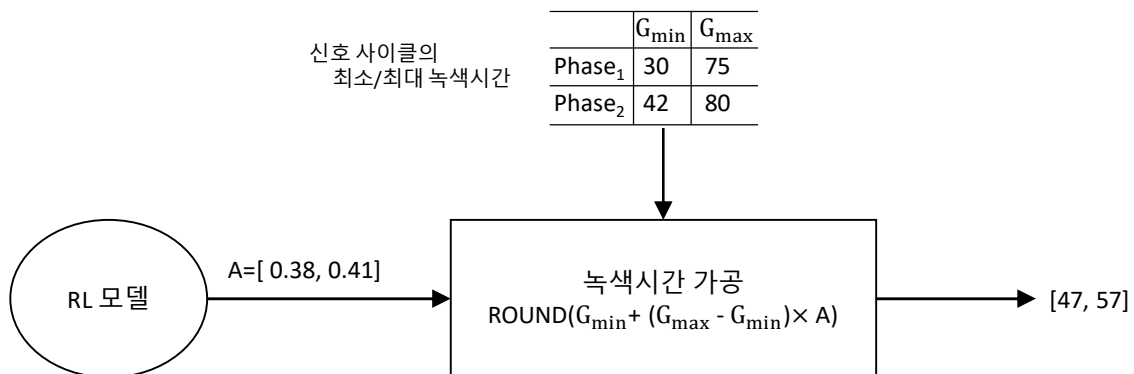
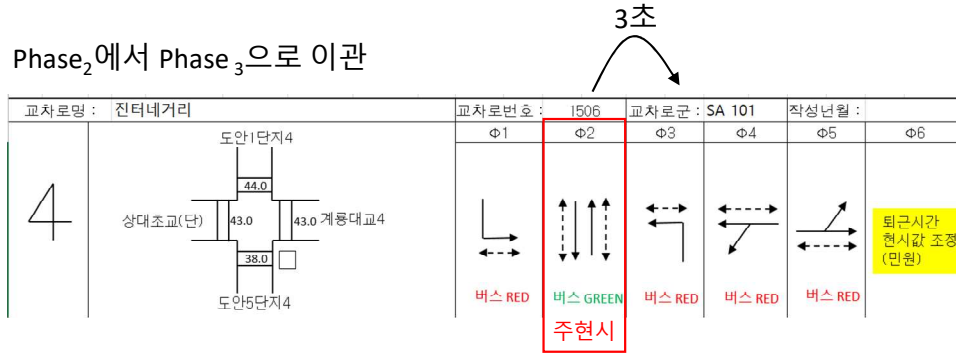


그림 8 모델 출력의 신호 사이클의 녹색시간으로 가공 예

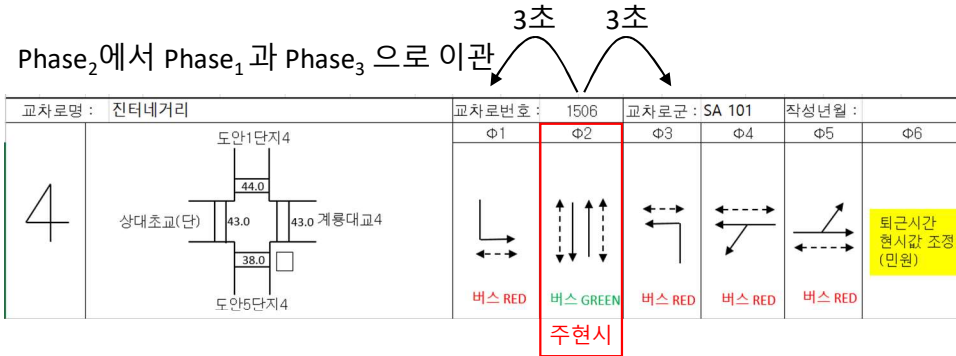
- (5) 주현시와 나머지 현시(하나 또는 여러 개)를 선택해서 녹색 시간을 일정 크기만큼 증감  
 지자체에서는 교통 소통 민원 발생시 신호 데이터 녹색시간을 주현시를 중심으로 나머지 현시를 T초씩 가감하여 조정하는 방법을 사용한다. 그림 9는 현실 제약 조건을 모두 만족시키는 신호 조정의 예이다. 지자체 신호 데이터 녹색시간 조정 방법을 참고하여 가능한 조정의 모든 경우를 나열하여 이를 현시 조정 테이블을 생성한다. 이때 최소 녹색시간과 최대 녹색시간 제약 조건을 활용하여 행동 공간을 줄인다.

- 1 강화학습 모델의 출력을 현시 조정 테이블 상의 엔트리 중 하나로 매핑하여 주현시와
- 2 나머지 현시들 간의 녹색 시간 조정으로 가공한다. 그림 10은 현시 조정 테이블을
- 3 활용한 모델 출력의 신호 사이클 녹색시간으로 가공하는 예이다.

Phase<sub>2</sub>에서 Phase<sub>3</sub>으로 이관

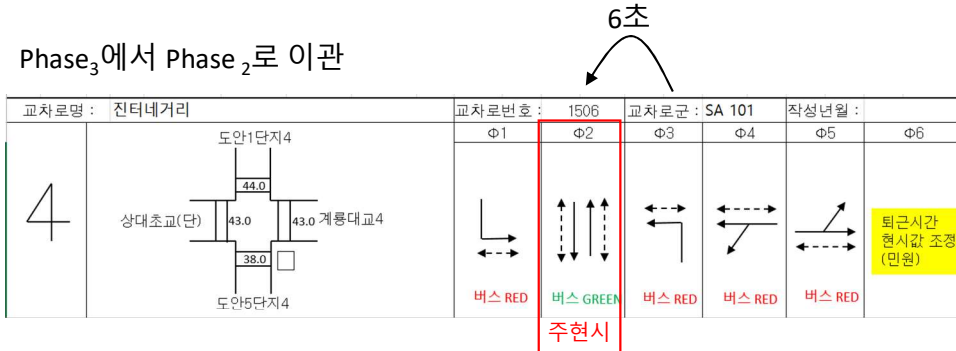


Phase<sub>2</sub>에서 Phase<sub>1</sub>과 Phase<sub>3</sub>으로 이관

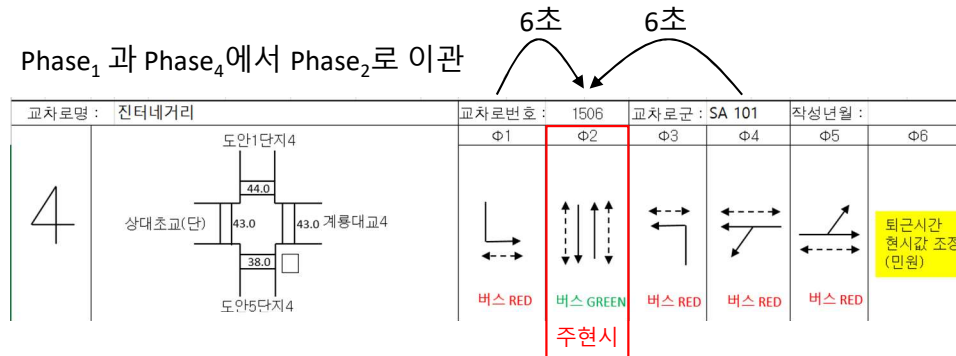


(가) 주현시에서 나머지 현시로 녹색시간 이관

Phase<sub>3</sub>에서 Phase<sub>2</sub>로 이관



Phase<sub>1</sub>과 Phase<sub>4</sub>에서 Phase<sub>2</sub>로 이관



(나) 나머지 현시에서 주현시로 녹색시간 이관

1

2

그림 9 지자체 신호 조정 방법에 기반한 현실 제약 조건을 모두 만족시키는 신호 조정 예

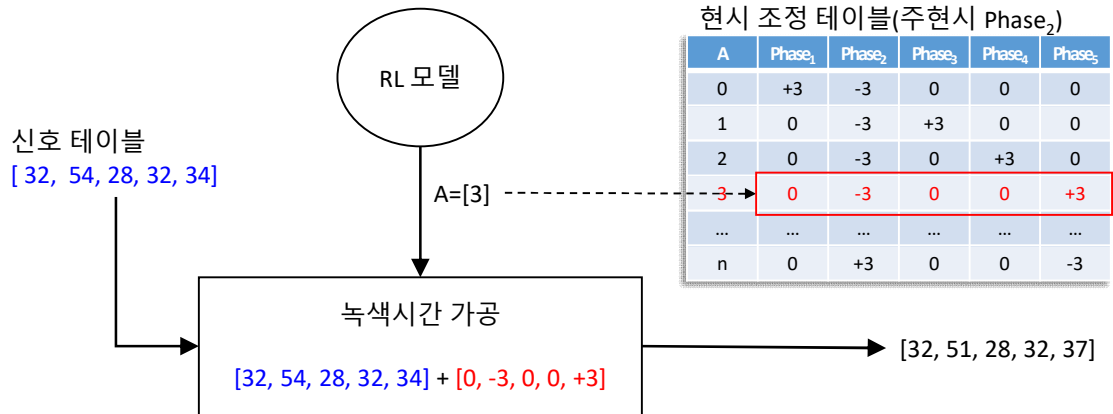


그림 10 현시 조정 테이블을 활용한 모델 출력의 신호사이클 녹색시간으로 가공 예

#### 4.1.5 교통 신호 최적화 강화학습을 위한 도커 이미지 제공

UNIQ 시스템에서는 클라우드 서비스를 활용한 도심 교통 혼잡 제어를 지원할 수 있어야 한다. 이를 위해 교통 신호 최적화를 위한 강화학습 환경을 도커 이미지화하여 클라우드 엣지 통합 관리 서브시스템에서 생성하는 도커 컨테이너에서 활용할 수 있도록 한다.

교통 신호 최적화를 위한 강화학습 도커 이미지에는 다음을 포함하도록 구성하여 생성한다.

- 교통 신호 최적화를 위한 가상 교통 환경인 교통 시뮬레이터 바이너리 : SALT
- 교통 시뮬레이터가 동작하기 위해 필요한 라이브러리 : 부트스트랩, gcc 등
- 교통 신호 최적화에 사용될 강화학습 학습 코드 : 정책(모델), 상태, 보상, 행동
- 강화학습 코드가 동작하기 위해 필요한 라이브러리 : 파이썬, 텐서플로우 등

생성한 도커 이미지는 도커 이미지 저장소인 도커허브(<https://hub.docker.com>)를 이용하여 클라우드 상에 저장하고, 필요시 다운로드하여 사용할 수 있게 한다.

## 4.2 대규모 교통 네트워크 신호 최적화 분산 처리

교통 신호 최적화 서브시스템에서는 대규모 교통 네트워크에 대해 빠르고 효과적인 교통 신호 최적화를 통해 교통 소통을 원활하게 할수 있도록 하기 위해 서로 영향을 주고받는 교차로들을 그룹핑하여 다수의 에이전트를 이용하여 분산 처리를 한다. 이를 위한 상세 기능 설계는 다음과 같다.

### 4.2.1 교차로 그룹핑

분산 처리를 통해 대규모 교통 네트워크에 대한 신호 최적화를 빠르고 효과적으로 하기 위해 서로 영향을 미치는 교차로들을 그룹핑한다

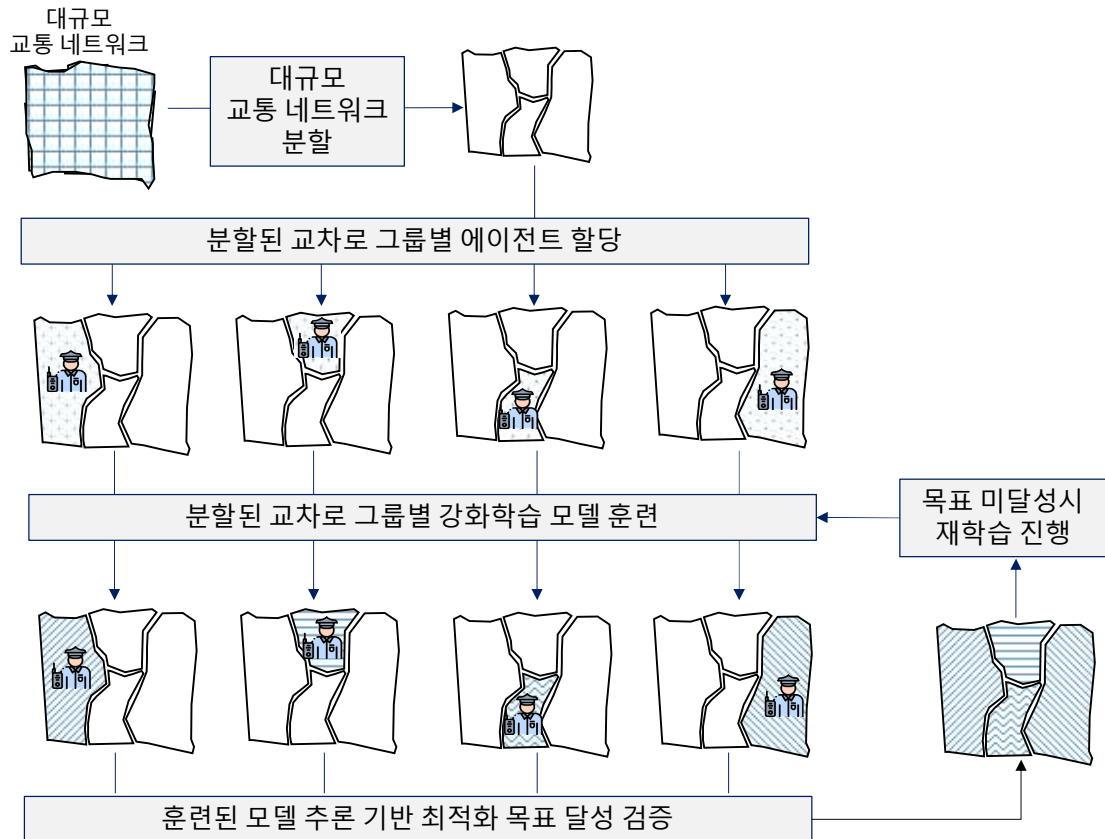
#### 가) 지자체의 신호 연동 교차로 그룹 활용

교통 소통을 원활히 하기 위해서는 신호 교차로에서 차량의 지체와 정지 수를 최소화하는 것이 중요하다. 이를 위해 특정 지역이나 신호 교차로가 연속적으로 위치한 간선도로 차원에서 상류부 신호 교차로에서 녹색신호를 받고 출발한 차량이 다음 교차로에서 정지하지 않고 그대로 통과하도록 신호 체계를 조정하는데 이를 신호 연동이라고 한다. 대부분의 지자체에서는 다년간의 교통 신호 운영 경험에 기반하여 서로 영향을 주고받는 교차로들을 그룹핑하고 그들간 신호 연동을 통해 교통 소통을 원활하게 하기위한 노력을 하고 있다. 이러한 지자체에서 신호 연동을 하고 있는 교차로 그룹을 신호 최적화를 위한 교차로 그룹으로 활용한다.

### 4.2.2 다중 에이전트 기반 신호 최적화

대규모 교통 네트워크를 구성하는 교차로들을 여러 교차로 그룹으로 분할한 후에 교차로 그룹별로 하나의 에이전트가 교통 신호 최적화를 담당하게 하는 다중 에이전트기반 분산 처리를 통해 대규모 교통 네트워크에 대한 최적화를 한다. 그림 11는 다중 에이전트 기반 신호 최적화를 그림으로 묘사한 것으로 다음과 같은 절차를 통해 교통 신호 최적화를 한다.





1  
2 그림 11 다중 에이전트기반 신호 최적화 개념

3 가) 분할된 교차로 그룹별 신호 최적화 에이전트 할당

4 하나의 노드에 신호 최적화를 담당할 여러 교차로 그룹을 할당하고, 각 노드에서는 할당된  
5 교차로 그룹별로 교통 혼잡을 개선하기위한 하나의 강화학습 기반 신호 최적화 에이전트를 할  
6 당한다. 신호 최적화 학습을 위한 경험의 확대를 위해 하나의 에이전트가 동시에 여러 모사 교  
7 통 환경을 활용하여 학습을 진행한다(그림 12 참고).

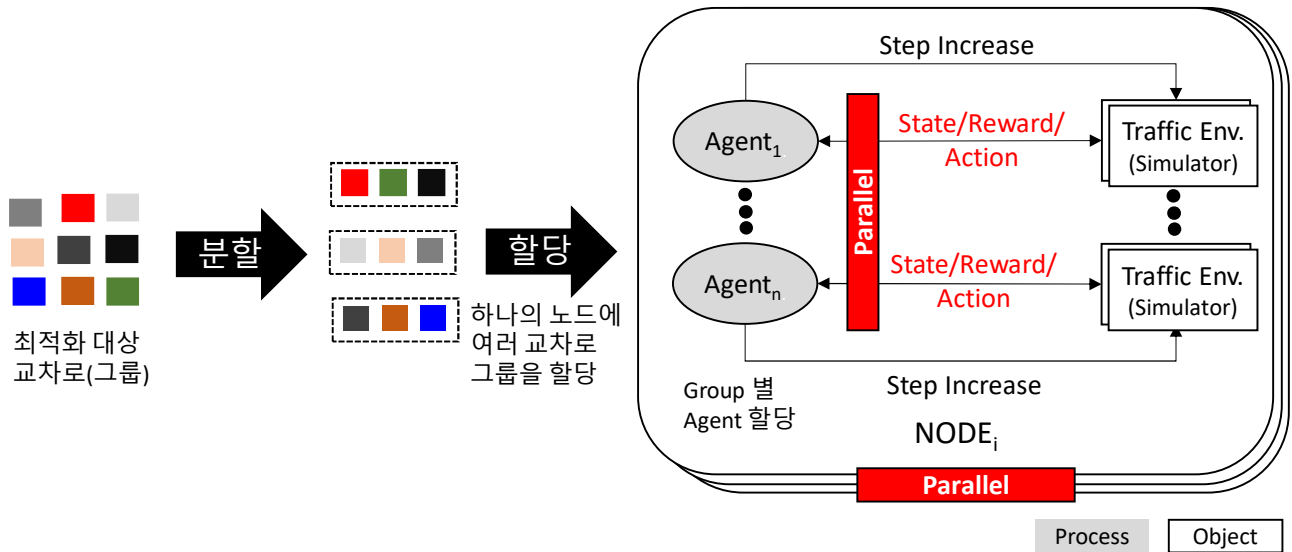


그림 12 교통 신호 최적화 에이전트 할당 개념

#### 나) 분할된 교차로 그룹별 신호 최적화를 위한 강화학습 모델 훈련

각각의 에이전트들은 자신이 담당하는 교차로 그룹의 교통 소통을 원활하게 하기위해 강화 학습에 기반하여 신호 최적화 모델 훈련을 통해 지역 신호 최적화를 수행한다. 이때 신호 최적화를 담당하도록 할당된 교차로 이외의 교차로는 최초 훈련시에는 고정신호에 기반하여 신호를 운영하고, 학습된 모델이 있는 경우에는 이들 모델을 활용한 추론에 기반하여 신호를 운영한다.

#### 다) 훈련된 모델 추론 기반 최적화 목표 달성 검증

지역 신호 최적화를 위해 훈련된 강화학습 모델들에 기반한 최적 신호 추론을 통해 신호 제어 후 전체 교통 네트워크 수준의 최적화 학습 목표를 달성했는지 검증한다. 이때 신호 최적화 학습 목표 달성 여부는 교차로 통과시간의 변화를 측정하여 활용한다.

#### 라) 목표 미달성시 그룹별 재학습 진행

만약 교차로 통과시간 감소 목표를 달성하지 못한 경우에는 나) 단계로 돌아가서 지역별로 재학습을 진행한다. 교차로 통과시간 감소 목표를 달성한 경우에는 모델의 추론 결과를 이용하여 신호를 도출하고 실세계 교차로의 신호 시스템으로 적용한다.

### 4.3 전이학습 기반 신호 최적화 강화학습 모델 생성

새로운 교차로에서의 학습 시간을 절약할 수 있도록, 교통 특성이 유사한 교차로 그룹에서 훈련된 모델을 재활용하는 전이 학습 기반의 신호 최적화 기능 설계는 다음과 같다(그림 13 참

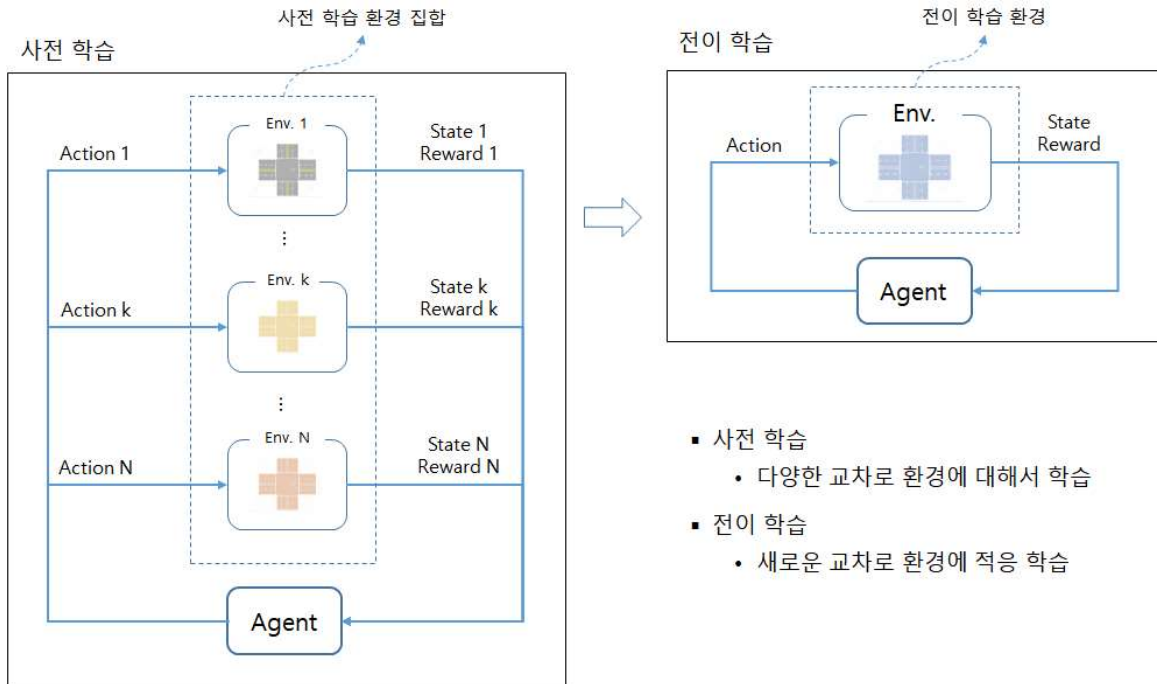


그림 13 전이학습 기반 신호 최적화 개념

#### 4.3.1 사전 학습 훈련 및 사전 학습 모델 생성

사전 학습은 최적 신호 제어를 위해 필요한 기본적인 행동과 전략을 학습하는 단계이다. 사전 학습 환경 집합에 있는 다양한 환경과 교통 상황에 대해 학습하여, 일반적인 교통 효율성 개선 능력을 갖춘 사전 학습 모델을 생성한다. 사전 학습 모델은 새로운 교차로 환경에 적용하여 재 활용된다.

#### 4.3.2 전이 학습 훈련 및 전이 학습 모델 생성

전이 학습에서는 사전 학습 모델을 새로운 교통 환경에 적용하여 추가적인 학습을 수행한다. 사전 학습 모델이 새로운 환경에 맞도록 조정되는 전이 학습을 거쳐, 새로운 교통 환경에 적합한 교통 신호 최적화 모델인 전이 학습 모델이 생성된다.

## 4.4 최적 설계기반 교통 신호 최적화

수집한 교통 데이터에 기반하여 교통류의 특성을 분석하고, 교통 공학을 활용한 시간대 별 교통 흐름 변화에 따른 신호 분할과 교차로 간 신호 연동 등을 통해 교통 소통을 원활하게 하기 위한 최적 설계기반 신호 최적화의 기능은 다음과 같다. 상세한 설계 내용은 별도의 문서로 작성된 “최적 설계기반 교통 신호 최적화 블록 설계서[6]”를 참고한다.

### 4.4.1 최적 신호 설계를 위한 교통 데이터 분석

교통 공학에 기반한 신호 현시 분할, 연동축 분할 등을 위해 사용될 수 있도록 계층적 교통 데이터 관리 시스템에 저장되어 있는 수집된 교통 데이터에 대한 시간대별, 회전 방향별 통행 비율 분석 및 주요 교차로에 대한 유입/유출 경로 분석 등의 분석을 한다.

### 4.4.2 단일 교차로에 대한 신호 현시 분할

분석된 단일 교차로에서의 시간대별/방향별 통행 비율과 전체 통행량 정보를 활용하여 K-평균 군집(K-means clustering)에 기반한 군집화를 통해 하루를 유사한 통행 특성을 갖는 시간 그룹으로 분할하는 신호 현시 분할을 한다(그림 14 참고).



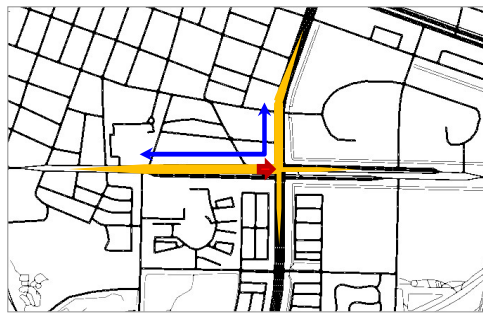
그림 14 신호 현시 분할 과정

### 4.4.3 다중 교차로에 대한 연동축 그룹핑

경로형 통행 데이터를 활용한 주요 교차로를 중심으로 유입/유출 경로 분석 결과에 기반하여 통행 빈도가 많은 연속된 구간을 연동축으로 분류 후 해당 교차로들의 신호를 그룹핑한다.

#### 4.4.4 연동 교차로 신호간 신호 오프셋 산출

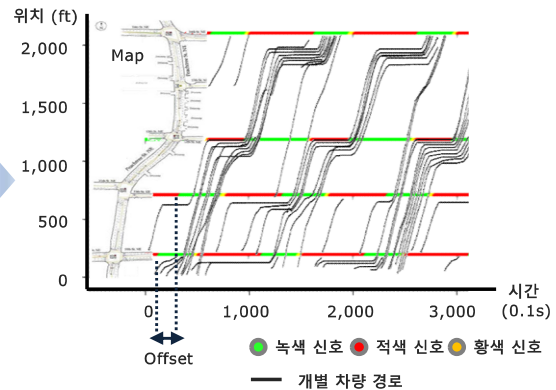
분류된 연동축마다 교차로간 거리, 적정 속도 기준 통과 소요시간 등을 고려하여 차량이 해당 연동축(신호 연동 교차로) 구간을 신호 지체없이 통과할 수 있도록 신호 그룹의 오프셋을 산정한다(그림 15 참고).



■ 해당 교차로 방향(→)으로 진입하는 차량들의 경로 밀도

□ 연동축 구간

연동 축 구간 선정



연동 오프셋 (Offset) 산출

그림 15 연동축 구간 선정 및 연동 오프셋 산출 개념도

## 5. 서브 시스템 인터페이스

### 5.1 교통 신호 최적화 실행 인터페이스

#### 5.1.1 교통 신호 최적화 실행 프로그램 직접 실행

가) 단일 노드 환경 및 전이 학습

강화학습 기반 신호 최적화 모델의 학습 혹은 학습된 모델을 활용한 최적 신호 추론을 위해서는 파이썬 파일(run.py) 실행을 활용한다. 아래는 그 예이다.

```
[%] python run_main.py --mode train --scenario doan --map doan --target-TL "SA 101"
    --mem-len 512 --action ga --reward-func wq --control-cycle 5 --epoch 2000 --
```

각각 인자에 대한 설명은 다음과 같다.

표 1 교통 신호 최적화 실행 인자: 단일 노드 환경

인자	설명
--mode	수행 모드 : {train, test, simulate} train : 강화학습 훈련 test : 훈련된 모델 평가, simulate : 고정 신호 활용 제어
--scenario-file-path	교통 시나리오가 위치한 최상위 경로(상대 경로)
--map	맵(시나리오) 이름 : {dj_all, dj_doan, sa28, sa_1_6_17, cdd1, cdd2, cdd3} dj_all : 대전 전지역 dj_doan : 도안 지역 sa28 : 유성, 월평 지역 sa_1_6_17 : 교차로그룹 SA 1, SA 6, SA 17 cdd1,cdd2,cdd3 : 후보지역 1, 2, 3
--target-TL	신호 최적화 대상 교차로 그룹으로, 그룹간 콤마(,)로 분리 (예. --target-TL 'SA 101, SA 104')
--start-time	시나리오 상 시뮬레이션 시작 시간, 초
--end-time	시나리오 상 시뮬레이션 종료 시간, 초

인자	설명
--method	최적화 방법: { sappo } sappo : 교차로 그룹별 최적화
--action	강화학습의 행동 : {kc, offset, gr, gro, gt, ga} kc : 신호 유지 혹은 변경(신호 순서 제약 만족) offset : 오프셋 조정, gr : 녹색 신호 지속시간 조정 gro : 녹색신호 지속시간과 오프셋 조정 gt: 최소/최대 시간 내에서 녹색 시간 조정 ga: 최소 시간 보장. 주기 내에서 나머지 시간을 배분.
--state	강화학습의 상태 : {v, d, vd, vdd} v : 차선별 통과 차량 수 d : 차선별 밀도, vd : 차선별 통과 차량 수와 밀도 vdd : 차선별 통과 차량수를 밀도로 나눈 값
--reward-func	강화학습의 보상 : {pn, wt, wq, tt, cwq} pn : 통과 차량 수 wt : 대기 시간 wq : 대기열 길이 tt : 여행 시간 cwq : 축적된 대기열 길이
--cumulative-training	이전에 학습된 모델을 활용하여 축적 학습을 할지 여부 : true 혹은 false
--model-num	평가할 모델 번호
--infer-model-num	학습 중 추론에 이용할 모델 번호 : infer-TL이 존재할때만 의미를 가짐
--result-comp	모델 훈련 결과 점검을 위해 고정신호에 기반한 제어와 비교 여부 : {true, false}
--io-home	실험을 위한 입출력 최상위 디렉토리로 상대 경로
--epoch	학습(훈련)을 진행할 에포크 수
--warmup-time	학습을 수행하지 않고 교통 상황 모사를 위한 워밍업 시간: 초
--model-save-period	학습(훈련) 중 모델 저장 주기
--action-t	녹색 시간 부여/조정 단위
--reward-info-collection-cycle	보상 계산을 위한 정보 수집 주기

인자	설명
--reward-gather-unit	보상 정보 수집 단위 : {env, sa, tl} env : 교통 환경 전체 sa : 교차로 그룹 단위 tl : 신호등 단위
--gamma	강화학습 모델의 gamma
--epsilon	탐험 비율 초기 값
--epsilon-min	탐험 비율 최소값
--epsilon-decay	탐험 비율 축소 비율
--epoch-exploration-decay	축적 학습시 탐험 비율 축소 비율(per epoch)
--ppo-epoch	PPO 에이전트의 모델 갱신 시 에포크 수(model fit epoch)
--ppo-eps	PPO 에이전트의 EPS
--_lambda	PPO 에이전트의 lambda
--a-lr	PPO 에이전트를 구성하는 Actor의 학습률
--c-lr	PPO 에이전트를 구성하는 Critic의 학습률
--network-size	모델의 네트워크 크기로 콤마(,)로 구별되는 숫자 문자열
--optimizer	PPO 모델의 옵티마이저 : {adadelat, adagrad, adam, adamaz, ftrl, nadam, rmsprop, sgd}
--actionp	신호 변경 유무가 action인 경우에 신호 유지 선택 비율
--mem-len	재현 메모리 크기
--mem-fr	재현 메모리 망각(삭제) 비율
--offset-range	오프셋 조정 범위 인자
--control-cycle	강화학습 에이전트에 의한 신호 변경 주기
--add-time	녹색 신호 지속 시간 조정 최소 단위
--infer-TL	모델 학습 중 이전 학습된 모델의 추론을 이용할 교차로 그룹으로 콤마(,)로 구분(ex. --infer_TL 'SA 101, SA 104')
--infer-model-path	추론에 사용할 모델이 저장된 경로
--num-of-optimal-model-candidate	최적 모델 선정시 보상을 비교할 후보 수

1

## 2 나) 다중 노드 환경(분산 처리)

3 대규모 교통 네트워크를 구성하는 교차로들에 대한 신호 최적화를 위해 분산 처리를 하는  
4 데, 다음의 절차를 이용하여 다중 노드 환경에서 분산 학습을 수행한다.

5 (1) 참여 노드들에 대한 패스워드없는 로그인 환경 구축



ssh-keygen과 ssh-copy-id 명령어를 이용하여 참여 노드들간에는 패스워드없이 접근 가능한 환경을 구축한다.

## (2) 분산 학습 제어 데몬 실행

제어 노드에서 전체 학습을 제어를 담당할 분산 학습 제어 데몬을 실행시킨다. 아래는 그 예이다.

```
[%] python DistCtrlDaemon.py --port 2727 --map doan --target "SA 101, SA 104"
      --num-of-learning-daemon 2 --action gr --validation-criteria 20.0
      --epoch 100 --model-save-period 5
```

대부분의 실행 인자들은 단일 노드 환경에서의 실행을 위한 인자들과 이름과 의미가 동일하다. 다중 노드 환경에서 분산 학습을 위한 분산 학습 제어 데몬 실행에서만 사용되는 인자들은 다음과 같다.

표 2 분산 학습을 위한 제어 데몬 실행 인자

인자	설명
--port	제어 데몬과 연결하기 위한 포트
--validation-criteria	학습을 통해 달성하려는 성능 향상률 목표
--num-of-learning-daemon	학습을 위한 실행 데몬 수
--model-store-root-path	학습된 모델이 저장될 경로

## (3) 분산 학습 학습 수행 데몬 실행

학습 수행을 담당할 노드들에서 분산 학습 학습 실행 데몬을 실행시킨다. 아래는 그 예이다.

```
[%] python DistExecDaemon.py --ip_addr 192.168.1.10 --port 2727
```

분산 학습 수행 데몬에서 사용되는 인자들은 다음과 같다.

표 3 분산 학습을 위한 제어 데몬 실행 인자

인자	설명
--ip-addr	연결할 제어 데몬의 IP 주소
--port	제어 데몬과 연결하기 위한 포트

### 5.1.2 클라우드 서비스를 위한 도커 이미지 활용

도시 교통 신호 최적화에 대한 클라우드 서비스 제공을 위해 도커 이미지를 활용하는 경우에는 도커 명령어를 직접 사용한다. 도커 명령행 인터페이스의 상세한 내용은 참고문헌 [5] 을 참고한다.

## 5.2 신호 현시 분할 제공 인터페이스

단일 교차로에 대해 교통량 데이터를 기반으로 신호 TOD를 분할하여 제공하기 위해서는 다음의 명령어를 활용한다.

```
python TOD.py --intersection_id id --start_date YYYY-MM-DD --end_date YYYY-MM-DD --max_tod n
```

--intersection\_id : TOD 분할할 교차로 식별자

--start\_date : 분석할 데이터 범위로 시작 시간 (YYYY-MM-DD)

--end\_date: 분석 데이터 범위로 종료 시간(YYYY-MM-DD)

--max\_tod: 고려 가능한 최대 TOD 개수

산정된 TOD 정보는 아래의 예처럼 csv 형식으로 전달되며, 산출된 TOD 정보에 대한 상세 규격은 추후 확정될 예정이다.

IXR_ID	DAY_DVS	ANLY_DT	TOD_HH	TOD_MM	CLCT_STRT_DT	CLCT_END_DT	TOD_NMBR	DVS
1865863300	0	20210913181131	0	0	202108010000	202108310000	0	0
1865863300	0	20210913181131	5	30	202108010000	202108310000	1	0
1865863300	0	20210913181131	6	30	202108010000	202108310000	2	0
1865863300	0	20210913181131	7	30	202108010000	202108310000	3	0
1865863300	0	20210913181131	9	0	202108010000	202108310000	4	0
1865863300	0	20210913181131	20	30	202108010000	202108310000	5	0
1865863300	0	20210913181131	22	30	202108010000	202108310000	6	0

그림 16 신호 TOD 분할 결과 예