

# 대규모 교통 네트워크에 대한 강화학습 기반 신호 제어 학습 상세 설계(안)

- 학습의 문제점과 해결
- 구조별 비교
- 세부 기능
- 할 일

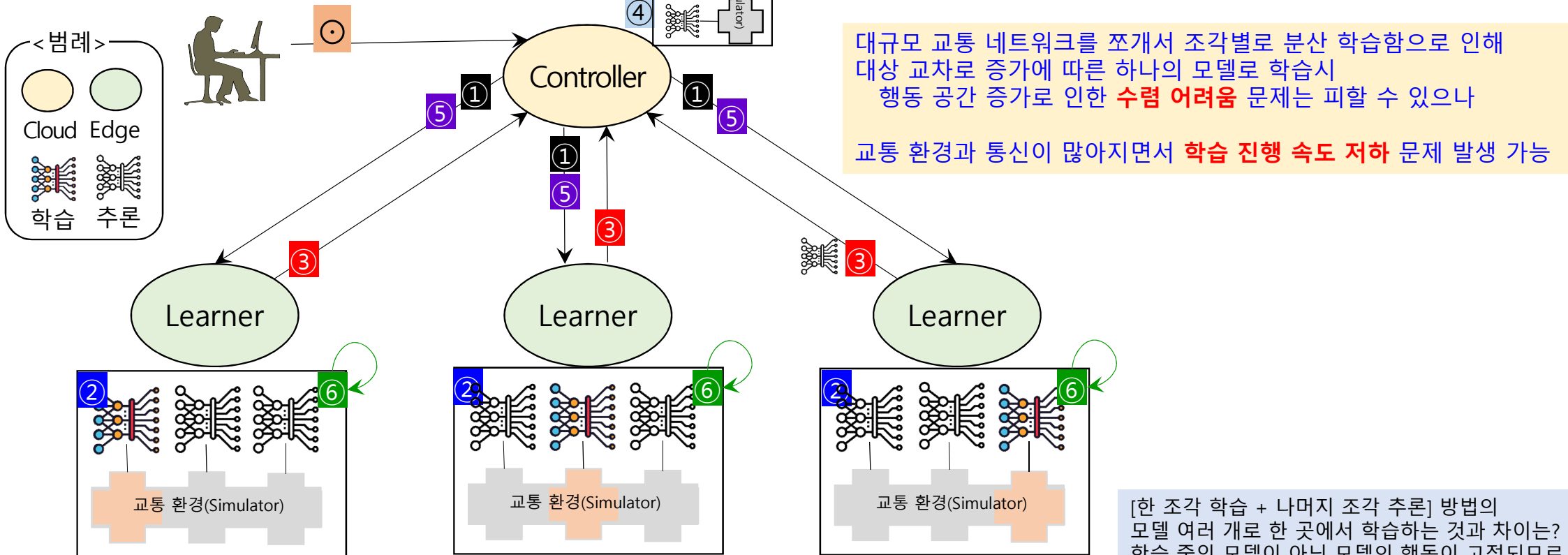
## 대규모 교통 네트워크에 대한 강화학습 기반 신호 제어 학습의 문제점 및 해결

- 학습 소요 시간이 길 것으로 예상되는데 회피할 수 없다.
  - (원인) 대규모 교통 네트워크에 대한 상태 정보 수집/제어를 위한 시뮬레이터와 통신 부하임.
  - 하나의 지역(교차로그룹) 학습 결과로 모델이 아닌 고정 신호를 만들어 줄수 있다면 통신 부하를 줄일수 있음

## 대규모 교통 네트워크에 대한 강화학습 기반 신호 제어 학습

- 수렴의 어려움
  - 제어 대상이 많아지면서 모델의 행동 공간이 크다
- 학습시간이 오래 걸림
  - 상태 정보 수집 교차로와 제어 교차로의 수가 증가함에 따라 교통 환경과 통신을 위한 시간 증가
- 해결 방안
  - 분할 학습 : 모델의 제어 대상 교차로 수를 줄여 행동 공간을 줄인다
  - 분산 학습 : 분할된 것들의 학습을 동시에 진행

# 구조 : 엣지-클라우드 협업 + 분산



- ① Controller에 감소율 목표를 포함하는 신호최적화 요청(설정) 정보를 전달한다.
- ② Controller에서는 각 Learner에 학습을 담당할 교차로 그룹을 알린다.
- ③ 각 Learner는 교통환경과 연동하여 담당 교차로그룹에 대한 모델 학습을 한다. 이때 다른 교차로 그룹은 초기화된 고정 신호로 동작한다.
- ④ 각 Learner에서는 담당하는 특정 교차로그룹에 대한 학습 결과인 모델을 Controller에 전송한다.
- ⑤ Controller에서는 각 Learner의 학습 결과를 활용하여 교통 환경과 연동하여 신호 최적화 목표를 달성했는지 확인을 위한 실험을 하여, 만약 신호 최적화 목표를 달성했으면 종료한다.
- ⑥ 신호최적화 목표를 달성하지 못했으면, Controller에서는 Learner로부터 받은 모든 교차로 그룹에 대한 모델들을 각 Learner들에게 전송한다.
- ⑦ Learner에서는 담당하는 그룹에 대해 학습을 다시 수행한다. 이때 다른 그룹은 Controller에서 받은 ML 모델 이용한 추론을 적용한다. 학습 완료 후 ③ 으로 간다.

Learner에서 학습된 모델과 함께 고정 신호를 Controller에 전달해 줄 수 있을까?  
학습시 대상 외 교차로는  
계속 고정 신호로 진행이 가능할 텐데...

## 대규모 교통 네트워크에 대한 강화학습 기반 신호 제어 학습

- 수렴의 어려움 → 분할로 해결 가능
- 학습 시간이 오래 걸리는 문제는 여전히 존재
  - 분산 학습에서 학습 대상이 아닌 교차로에 대해 추론 이용하여 제어
  - 분산 학습에서 학습/추론을 하는 경우에는 상태 정보 수집과 제어 위해 교통 환경과 통신 필요
- 상태 정보 수집 회수를 줄이는 방법은?
  - 학습된 모델에서 고정 신호 정보를 얻어올 수 있을까?
    - If possible
      - 학습부에서 학습된 모델과 함께 고정 신호 전송
      - 검증에는 모델 추론 이용하게 하고, 다른 모델 학습시에는 고정 신호 이용하게

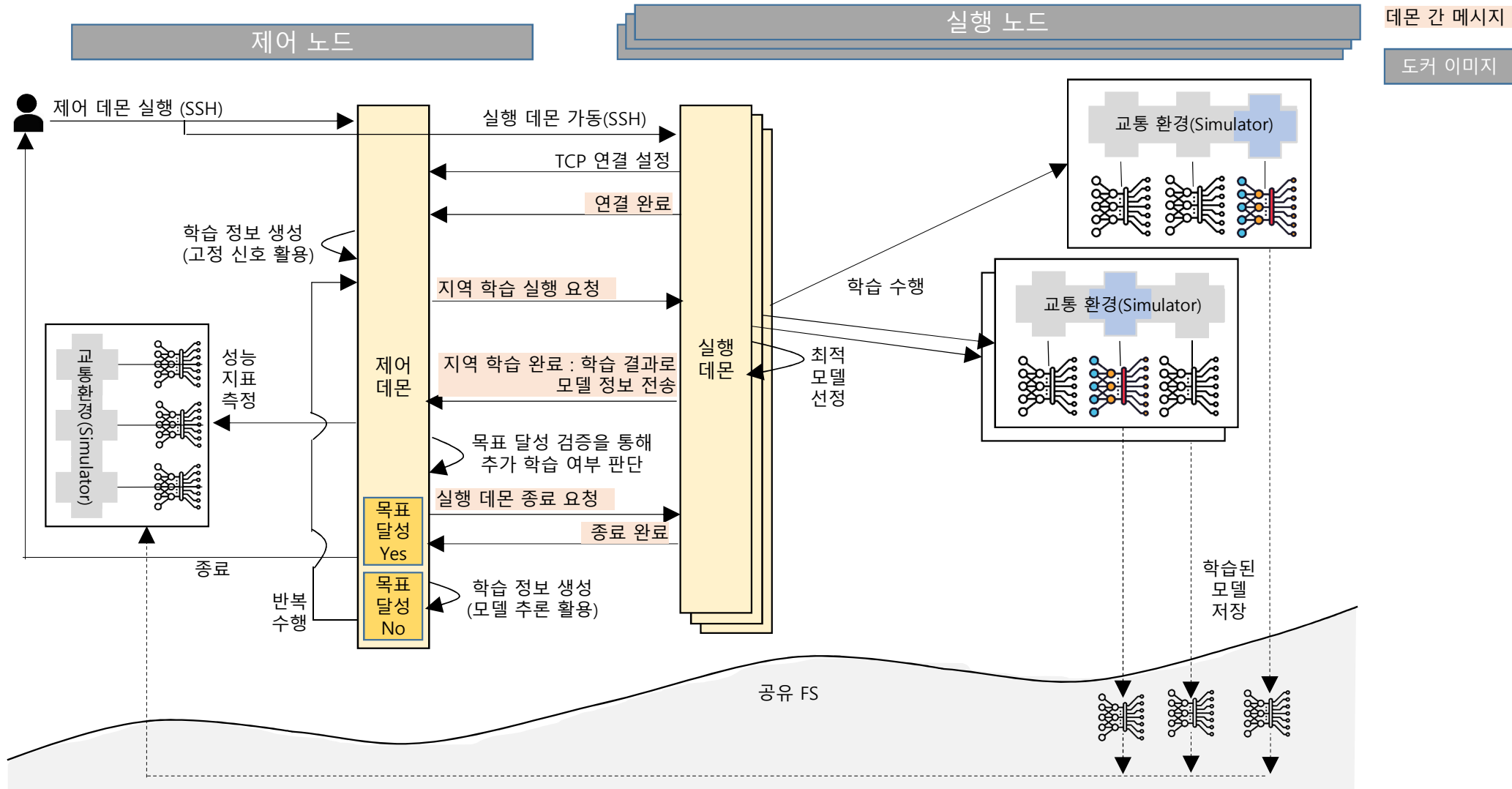
## 참고 : 강화학습기반 신호 제어를 위한 구조별 비교

구조		모사 교통 환경 개수	모델 수	상태 정보 접근 교차로	비고
단일 학습	하나의 모델로 전체 제어 학습	하나	1	전체 교차로	행동 공간이 커서 수렴 어려움
	여러 모델로 한 곳에서 전체 학습	하나	N	전체 교차로	모든 모델이 학습으로 계속 진화됨으로 인해 모든 교차로의 제어 행동이 동일한 상태 입력에도 고정되지 않음
분산 학습	다수 노드 활용 학습 (학습 + 고정) <sup>1</sup>	여러 개	N	학습 대상 교차로	고정 신호 위해서는 학습된 모델로부터 고정 신호 얻을 수 있을까?
	다수 노드 활용 (학습 + 추론) <sup>2</sup>	여러 개	N	전체 교차로	학습 중이 아닌 교차로에 대한 모델의 행동이 입력에 따라 고정 → 불확실성이 줄어들음
검증	하나의 노드에서 전체 추론	하나	1 or N	전체 교차로	학습된 모델을 이용한 전체 교통 제어

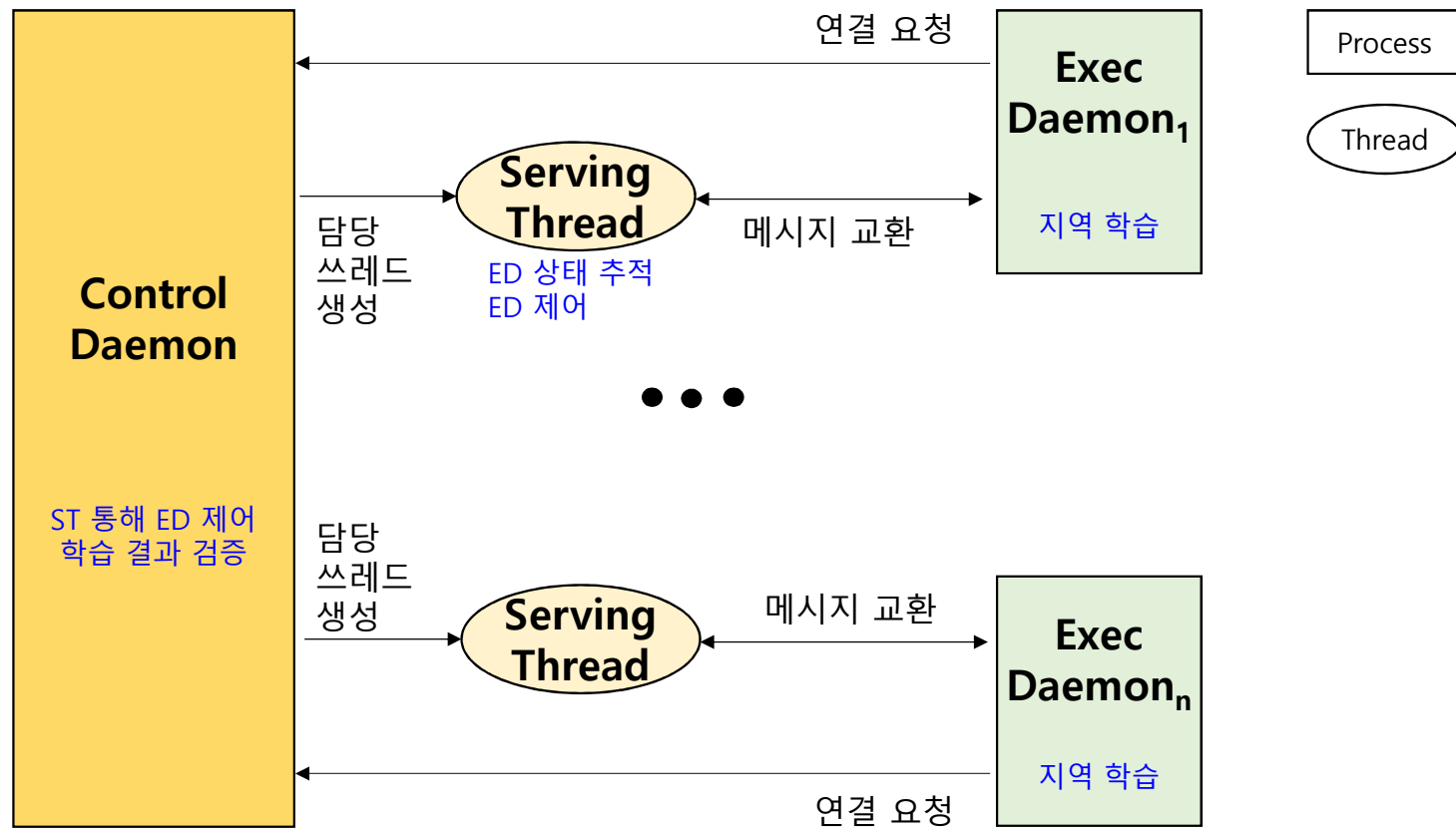
1 학습 + 고정 : 각 노드에서 담당하는 교차로에 대해서는 RL 학습하고 이외의 교차로는 고정 신호 이용 신호 제어

2 학습 + 추론 : 각 노드에서 담당하는 교차로에 대해서는 RL 학습하고 이외의 교차로는 RL 추론 이용 신호 제어

# 세부 기능 : 동작 흐름



## 세부 기능 : 프로세스 구조 및 기능





## 개발 관점에서 할 일....

- 가정
  - 강화학습 기반 신호 최적화를 위한 환경이 설정된 도커 이미지 존재
  - 하나의 그룹에 대한 강화학습 기반 신호 최적화 프로그램 존재
- (클라우드 활용) 시스템 환경 구성
  - 공유 가능한 디렉토리 설정
  - 공유 디렉토리를 도커 인스턴스에 연결
  - 도커 이미지 인스턴스 간 통신이 가능하도록 설정
  - 학습을 위한 도커 이미지 활용 클러스터 구성 : 제어 1 + 학습 n
  - 클러스터 내 노드간 비밀번호 없이 로그인 가능하도록 ssh 설정
  - 각 노드(도커 이미지)에 (제어/실행)데몬 프로그램 실행할 수 있도록 설정
- (분산 처리) 데몬 개발
  - 제어 데몬과 학습 데몬 개발
    - 도커 이미지에서 실행될수 있도록 생성
    - 제어부 데몬과 실행 데몬이 메시지 교환하며 학습 진행
    - 공유해야할 데이터는 공유 FS로 이용
  - 교차로 그룹 분할 : 기존 교차로 그룹 이용
  - 제어 연산 개발
  - 실행 연산 개발

# 데몬의 연산

- 제어 데몬
  - 실행 데몬 가동
    - 제어 데몬 가동 스크립트에서 SSH 이용하여 실행 데몬 가동
  - 교차로 그룹 분할
    - 기존 교차로 그룹 이용
    - 사용자가 입력
    - 자동화(혼잡 정보 기반 교차로 그룹 분할)는 고려하지 않음
  - 제어 데몬과 실행 데몬간 메시지 교환
    - 메시지 정의 : 별도 페이지 참고
  - 학습 정보 생성
    - ~~• 교차로별 신호 제어 방법, 교통 시뮬레이션 시나리오 파일~~
      - ~~• 교차로별 신호 제어 방법 : 교차로명, 학습/추론/고정, 모델 위치~~
    - 전체 학습 수행 교차로, 학습 담당 교차로, 모델 위치, 교통 시뮬레이션 시나리오 파일
    - DIC 파일로 생성하여 정해진 장소에 넣어 놓는다.
  - 성능 지표 측정 : 목표 달성 검증
    - 학습된 모델의 추론에 기반한 신호 제어를 적용하여 성능 지표 측정하여 목표 달성 검증
- 실행 데몬
  - 학습 수행
    - 학습 정보를 이용하여 담당 교차로들에 대한 학습 수행
  - 최적 모델 선정
    - 가장 좋은 모델이 어떤 것인지 찾아서 선정
      - 기준은? 학습 목표와 동일한 항목

## 데몬 간 교환 메시지

- 연결 완료 : 실행 데몬 → 제어 데몬
  - 제어 데몬 : 연결된 실행 데몬인지 확인 후 제어 리스트에 추가
  - MSG\_CONNECT\_OK
- 지역 학습 실행 요청 : 제어 데몬 → 실행 데몬
  - 제어 데몬 : 학습 수행을 위한 정보를 메시지에 실어서 전송
  - MSG\_LOCAL\_LEARNING\_REQUEST
  - (body) 강화학습 설정 파일 위치
    - 전체 학습 수행 교차로, 학습 담당 교차로, 모델 위치, 교통 시뮬레이션 시나리오 정보
- 지역 학습 완료 : 실행 데몬 → 제어 데몬
  - 실행 데몬 : 메시지에 학습 결과 모델 정보를 메시지에 실어서 전송
  - MSG\_LOCAL\_LEARNING\_DONE
  - (body) 학습된 모델 위치
- 실행 데몬 종료 요청 : 제어 데몬 → 실행 데몬
  - 메시지를 수신한 실행 데몬 종료
  - MSG\_TERMINATE\_REQUEST
- 데몬 종료 완료 : 실행 데몬 --> 제어 데몬
  - 메시지를 수신한 제어 데몬은 제어 리스트에서 제거, TCP 연결 종료
  - MSG\_TERMINATE\_DONE

# 개발 방법

- 파이썬 통신 라이브러리와 셸 스크립트 활용 개발
  - 파이썬 : 제어 데몬, 실행 데몬
  - 셸 스크립트 : 데몬 관리(실행, 종료)
- 점진적 확장 개발
  - 흐름 검증
    - 데몬 통한 연결 및 메시지 교환
      - 왼쪽 흐름 참고
    - 메시지에 해당하는 행동을 간단한 연산에 매핑하여 단순화
      - 강화학습 수행 : 1~10 사이의 난수를 발생시키는 연산
      - 학습된 모델 전달 : 저장소의 특정 위치에 난수 저장
      - 검증 : 모든 실행기 전달받은 난수의 평균이 7이상
    - Python program 만으로 1차 개발
  - 교통 환경 연동 개발
    - 2개 교차로 학습 분산 처리
    - 2개 교차로군 학습 분산 처리
  - 클라우드 연결
    - 이노그리드 관리 테스트베드 이용
    - 파이썬 프로그램을 파이썬 + 셸 스크립트로 분할
    - Docker image 연동

