

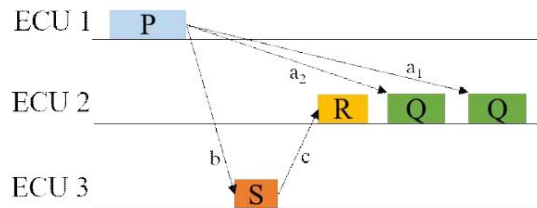
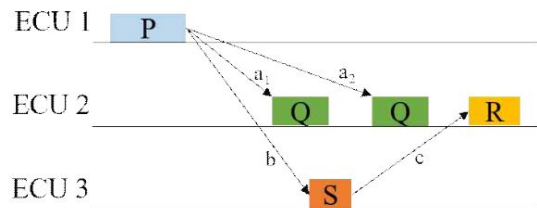
실시간 네트워크의 실행 모델 정의

2019. 5. 3. Show & Tell

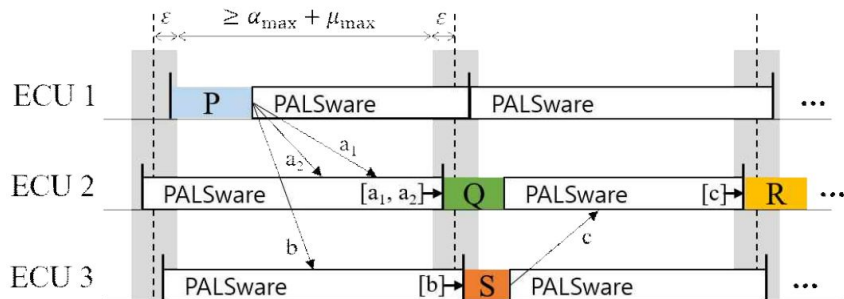
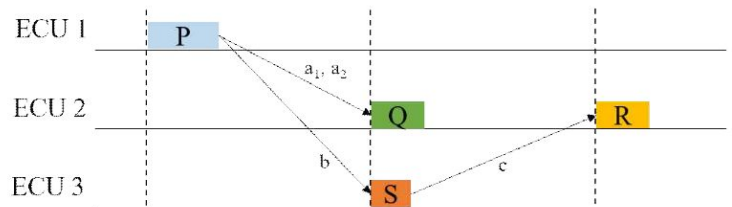
김윤승

동기식 · 비동기식 시스템 비교

- 비동기식: 무수한 실행의 가짓수



- 동기식: 실행의 가짓수를 줄임, 복잡성 감소



미들웨어 검증의 기대 효과

- 안전우선 시스템 - 완전한 검증이 목표
- 부품별 검증: 미들웨어 / 애플리케이션
- 애플리케이션의 디자인/검증을 더 쉽게 함

지난 시간에 이어..

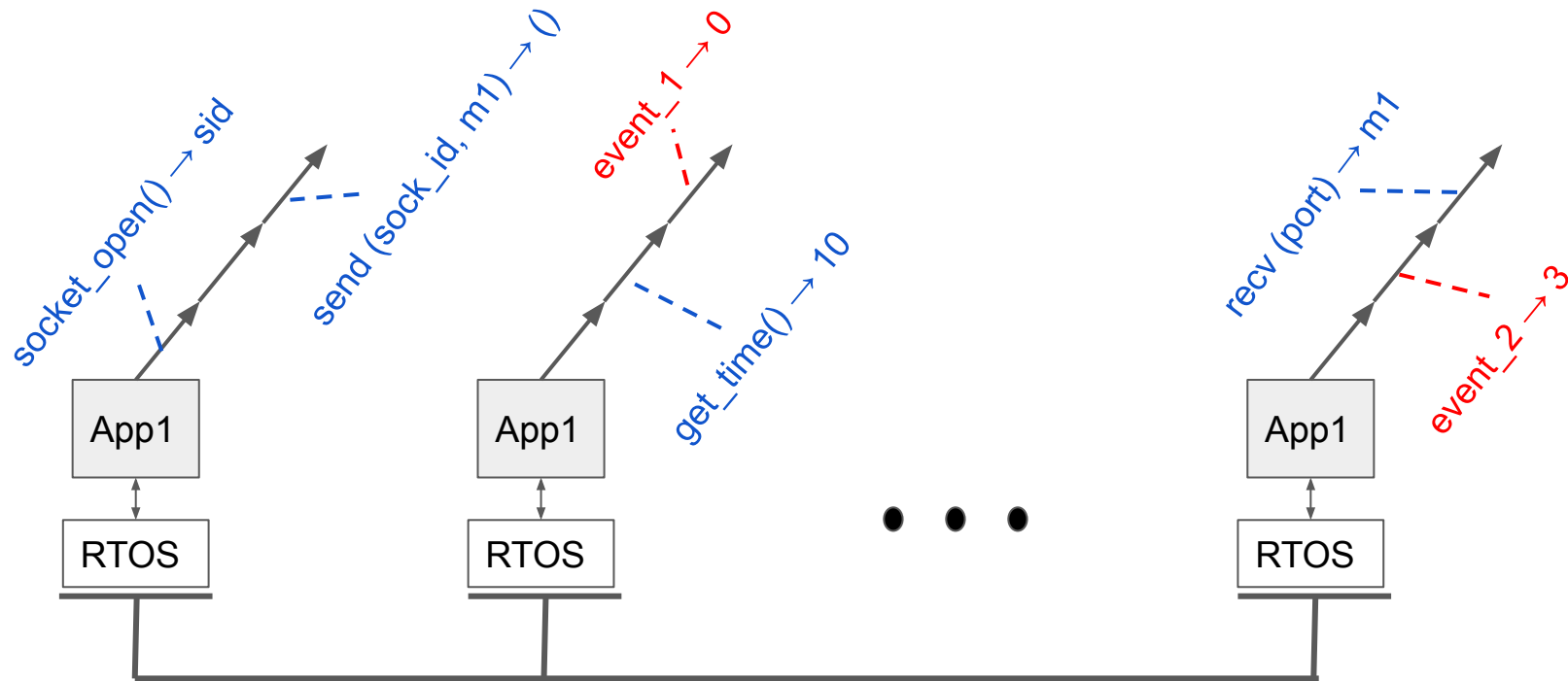
결론

- 프로그램의 행동을 interaction tree 형태로 정의

다음 계획

- 비동기식 네트워크-OS 모델 정의 완성
- 미들웨어 코드 + (모르는) 애플리케이션 코드 문제 해결
- [비동기식 하드웨어 실행 모델 행동 \subseteq 이상적 동기식 실행 모델 행동] 증명

전체 시스템 모델 (대략적인 형태)



애플리케이션 모델

$S_{site} : Set$

$_ \rightarrow _ : P(S_{site} \times SiteEvt \times S_{site})$

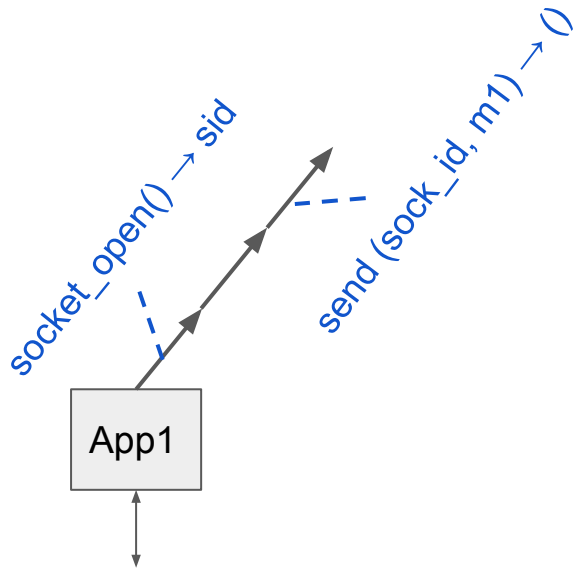
$init_state : P(S_{site})$

$final_state : P(S_{site})$

$SiteEvt := SysEvt \uplus ExtEvt$

$SysEvt := SendEvt \uplus RecvEvt \uplus \dots$

$SendEvt := (int \times Msg) \times Bool$



운영체제 모델

$$S_{os} := Timers \times Sockets$$

$$Time := Nat$$

$$t, t' : Time$$

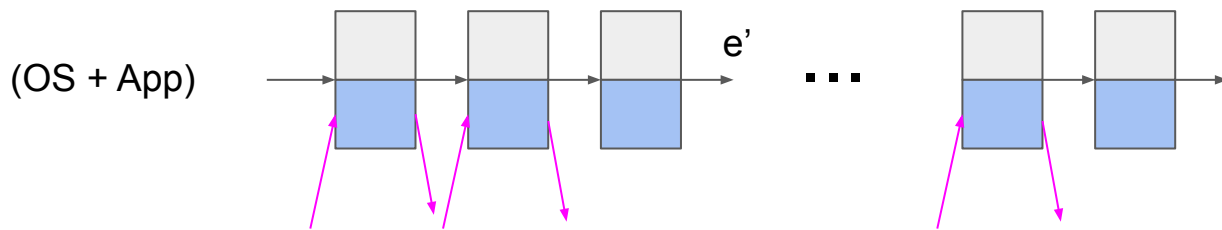
$$s_{os}, s'_{os} : S_{os}$$

$$e : SiteEvt = SysEvt \uplus ExtEvt$$

$$e' : ExtEvt$$

$$(t, s_{os}) \xrightarrow[e']{e} (s'_{os}, t') := ..$$

앱 + 운영체제 모델



메시지들

네트워크 모델

$$S_{nw} := list (Time \times Msg)$$

$$Msg := IP \times Port \times IP \times Bytes$$

$$(-, -) \rightarrow (-, -) : P(list(Msg) \times S_{nw} \times S_{nw} \times list(Msg)) := ..$$

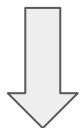
다음 계획

- 각 장소의 애플리케이션 실행 기계를 정의
 - C 프로그램(미들웨어 코드)과 임의의 기계(미들웨어의 app)가 함께 실행되도록 정의
- 동기식 모델 정의
- Interaction tree를 이용해, 실제 모델과 동기식 모델 사이의 증명 설계하기

사이트 + 운영체제 모델

$$s_{site} \rightarrow_e s'_{site}$$

$$(t, s_{os}) \xrightarrow{e}_{e'} (s'_{os}, t')$$



$$(t, s_{os}, s_{site}, \dots) \xrightarrow{e'} (t + 1, s'_{os}, s'_{site}, \dots)$$

시스템 모델