다양한 혼잡도에 적응하는 DQN 기반 Cellular V2X 분산혼잡제어

이은화, 정구선, *최주영, 문철
한국교통대학교, *연세대학교
20028606@ut.ac.kr, rntjs300@ut.ac.kr, cjy0103@yonsei.ac.kr, chmun@ut.ac.kr

Deep Q-Network Based Adaptive Distributed Congestion Control in Cellular V2X

Eun-Hwa Lee, Gu-Sun Joung, Joo-Young Choi*, Cheol Mun
Korea National University of Transportation, *Yonsei University

요 약

변동성이 있는 혼잡한 도로 상황에 따른 C-V2X(cellular vehicle-to-everything) 통신 시스템에서 패킷 충돌로 인한 패킷 정보 손실을 해결하기 위해 강화학습을 통해 차량마다 최적의 TTI(transmission time interval)를 조절하는 분산 혼잡제어 알고리즘을 제안한다. 제안된 기법이 기존 알고리즘보다 PDR(packet delivery ratio), IPG(inter-packet gap) 측면에서 높은 성능을 보여주는 것을 시뮬레이션 결과로 입증하고 다양한 혼잡도에서 학습한 결과와 단일 혼잡도에서 학습한 결과를 비교한다.

Ⅰ. 서 론

도로 환경의 안전과 관련하여 트래픽을 제어하기 위해 3GPP는 Release 14에서 LTE(long term evolution) C-V2X(cellular vehicle-to-everything) mode 4를 표준화하였다. mode 4는 셀룰러 인프라의 개입 없이 차량들이 SB-SPS(sensing based semi-persistent scheduling)를 통해 직접 무선자원을 할당하고 패킷을 전송한다. 하지만 혼잡한 도로 상황에서는 다수의 VUE(vehicular user equipment)가 동일한 무선자원을 예약하여 패킷이 충돌에 의해 손실되기 때문에, 이를 극복하기 위해 DCC(distributed congestion control)를 통해 전송 주기 TTI(transmission time interval)와 전송전력을 조절한다[1]. 아직 C-V2X에서 표준화된 DCC 알고리즘은 존재하지 않으나 IEEE 802.11p용 DCC표준인 SAE(society of automotive engineers)J2945/1[2]을 C-V2X에 적용하는 SAE J3161/1[3]을 개발하고 있다.

본 논문은 기존 SAE J2945/1의 전송 주기 제어 알고리즘을 강화학습 기법의 하나인 DQN(Deep Q-Networks)[5]을 이용하여 최적의 TTI를 결정하는 알고리즘을 제안한다. 또한 차량 밀도가 변하는 상황에서도 차량 밀도에 최적화된 TTI를 결정할 수 있는 다양한 차량 혼잡도에서 학습한 DQN 기반 분산 혼잡제어 알고리즘을 제안하고 이의 성능을 비교 분석하였다.

Ⅱ. 차량밀도에 적응하는 전송주기 제어 알고리즘

SAE J2945/1의 전송주기 제어 알고리즘은 시간 t에서 VUE i의 반경 100 m 내의 이웃 차량밀도 VD_t^i를 밀도계수 B와 비교하여 전송주기 TTI를 다음과 같이 결정한다[2].

$$TTI_t^i = \begin{cases} 100\ ms & VD_t^i \le B, \\ (100 \cdot VD_t^i)/B\ ms & B < VD_t^i < 6B, \\ 600\ ms & 6B \le VD_t^i \end{cases} \quad (1)$$

여기서, SAE J2945/1은 밀도계수 B를 25로 고정하므로, VD_t^i가 25 이상이 되면 혼잡도에 따라 TTI를 100ms에서 600ms까지 증가시키게 된다. 따라서, 밀도계수 B는 TTI를 조절해야 하는 혼잡구간과 최적 TTI 값을 결정하여 전송주기 제어 알고리즘의 성능을 좌우한다.

본 논문에서는 강화학습을 통해 다양한 차량 밀도에 따라 밀도계수 B를 최적화함으로써, 다양한 차량 밀도 환경에 적응하여 최적 TTI를 결정하는 분산 혼잡제어를 제안한다. 최적 B를 결정하기 위한 최적화 문제를 공식화하면 (2) 와 같다. 여기서, PDR(packet delivery ratio)을 최대화하되 CBR(channel busy ratio)을 target CBR에 근접하도록 한다. target CBR은 3GPP working document[4]와 같이 0.65로 설정하였다.

$$\mathbb{P}1: \max_{TTI_t^i} PDR_i[t,j] = \frac{\sum_{k\in\delta} C_{i,k}[t,j]}{\sum_{k\in\psi} F_{i,k}[t,j]} \quad s.t.\ \min_{TTI_t^i} \left| CBR^{target} - CBR_i[t] \right| \quad (2)$$

여기서, VUE i는 패킷을 송신하는 차량이며 ψ는 VUE i의 통신 범위에 있는 차량들의 집합이고 δ는 VUE i의 통신 범위에 있으며 패킷을 전송받은 차량들의 집합이다. VUE k는 δ와 ψ에 모두 포함되어 있어야 한다. $PDR_i[t,j]$은 서브프레임 t의 j번째 서브채널을 사용한 VUE i에서 오류없이 전송된 패킷 $C_{i,k}[t,j]$를 VUE i에서 송신한 총 패킷의 수인 $F_{i,k}[t,j]$로 나눈 것이며 VUE i의 패킷 성공률 이다. 식(2)는 $PDR_i[t,j]$을 최대화하며 CBR^{target}과 서브프레임 t에서 측정된 $CBR_i[t]$의 차이를 최소화한다.

Ⅲ. DQN 기반 혼잡도에 적응하는 전송주기 제어 알고리즘

DQN은 Q-learning을 deep neural networks로 근사화하여 Q-value를 구한다[5]. 네트워크 구조는 각 3개의 fully connected layer, ReLU activation layer와 action 개수만큼의 노드를 가진 output layer로 이루어져 있다. 그림 1과 같이 현재 time step t에서의 채널 이용률 $CBR_i[t]$을 state s_t^i로 네트워크의 입력단에 넣어서 출력으로 나온 Q-value들을 decaying ϵ-greedy를 이용해 exploration exploitation trade-off를 해주며 action a_t^i을 구한다. a_t^i는 B에 대한 이산값 {5,15,25,45,65}중에 하나이다.

에이전트의 리워드는 식 (2)를 기반으로 다음과 같이 결정된다.

$$R_j^t = PDR_i[t,j] + \left| CBR^{target} - CBR_i[t] \right| \quad (3)$$

리워드는 각 VUE가 DQN 에이전트가 되어 학습하는 과정에서 state에 따라 최적 action을 취할 수 있게 하는 지표로 사용된다. CTDE(centralized training decentralized execution)를 기반으로 하여 offline 학습을 하기 때문에 $PDR_i[t,j]$를 사용할 수 있다. 학습을 통하여 $(s_t^i, a_t^i, r_t^i, s_{t+1}^i)$ transition data가 완성되면 experience replay memory에 쌓이게 되고 이는 main network를 업데이트하는 minibatch data로 이용된다. batch size는 256이다. batch size만큼의 transition data를 통해 target network로 $\max_{a'}\widehat{Q}(s',a';\theta^-)$를 구하고 main network로 $Q(s,a;\theta)$를 구하여 아래의 식과 같이 Q-values의 MSE(mean square error) loss $L(\theta_t)$를 구한다.

$$L(\theta_t) = \mathbb{E}[(r(s,a)+\gamma \max_{a'}\widehat{Q}(s',a';\theta^-) - Q(s,a;\theta))^2] \quad (4)$$

위 식의 θ는 main network 파라미터로 (4)에 대한 gradient를 구하여 업데이트하고 θ^- target network 파라미터는 일정한 주기마다 main network와 동기화하여 안정적인 학습을 하도록 한다[5].

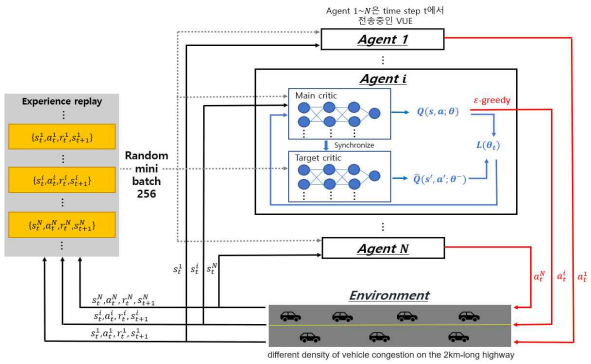

그림 1. 시뮬레이션에서 DQN 에이전트의 DCC 학습과정

Ⅳ. 시뮬레이션 결과

시뮬레이션은 LTEV2Vsim[6]을 기반으로 수행했다. LTEV2Vsim은 C-V2X 네트워크의 자원할당을 평가하기 위해 작성된 동적 시뮬레이터이다. 시뮬레이션의 환경 및 파라미터는 다음 표1과 같다.

표준 버전	C-V2X mode4	**MCS index**	6(=QPSK, 0.48)
시뮬레이션시간	20초	**패킷 크기**	300 bytes
중심 주파수	5.9 GHz	**SPS 임계값**	-94 dBm
대역폭	10 MHz	**전파모델**	WINNER Ⅱ, B1
도로 환경	2km 고속도로, 왕복 8차선		
차량 속도	60km/h, 표준편차 3km/h		
차량 밀도	2km 내 400, 600대		

표 1. 시뮬레이션 환경 및 시스템 파라미터

제안하는 알고리즘의 성능을 PDR과 평균 IPG(inter-packet gap) 관점에서 분석한다. 여기서 IPG는 동일한 송신 VUE와 수신 VUE 사이에서 연속적으로 오류 없이 수신된 2개의 패킷 사이 시간 간격이며, 평균 IPG는 송수신기 거리에 따른 시뮬레이션 환경 내 모든 VUE의 IPG에 대한 평균이다. 그림에서 fixed DQN은 고정된 하나의 차량 밀도에서 학습한 알고리즘이고 adaptive DQN은 다양한 차량 밀도에서 학습한 알고리즘이다.

그림 2는 2km 내 400대의 차량 밀도에서 학습한 fixed DQN과 400대와 600대의 차량 밀도에서 학습한 adaptive DQN 그리고 기존의 SAE J2945/1 알고리즘의 성능을 600대의 차량 밀도에서 테스트한 성능을 비교한 것이다. 제안하는 adaptive DQN은 기존 SAE J2945/1보다 PDR이 3% 개선되고 IPG는 10ms 정도의 미미한 증가에 그친다. 반면에 fixed DQN은 400대의 차량 밀도에 최적화된 과도하게 낮은 값의 TTI를 사용함으로써, 기존 알고리즘보다 PDR 성능이 최대 7% 감소하고 IPG가 과도하게 감소하는 결과를 보여준다. 그림 3은 2km내 600대의 차량 밀도에서 학습한 fixed DQN과 adaptive DQN 그리고 기존 알고리즘을 400대 차량 밀도에서 테스트한 성능을 비교한 것이다. adaptive DQN은 기존 SAE J2945/1에 비해 PDR이 개선되지만, 단일 혼잡도에서 학습을 수행한 fixed DQN은 600대의 차량 밀도에 최적화된 과도하게 큰 값의 TTI를 사용함으로써, PDR은 개선되나 IPG가 과도하게 증가하는 현상을 보여준다. 제안하는 다양한 차량 혼잡도에서 학습한 DQN 기반 분산 혼잡제어 알고리즘은 차량 밀도가 변하는 상황에서도 차량 밀도에 최적화된 TTI를 결정함으로써 기존 SAE J2945/1보다 우수한 PDR 성능을 보임을 확인할 수 있다.

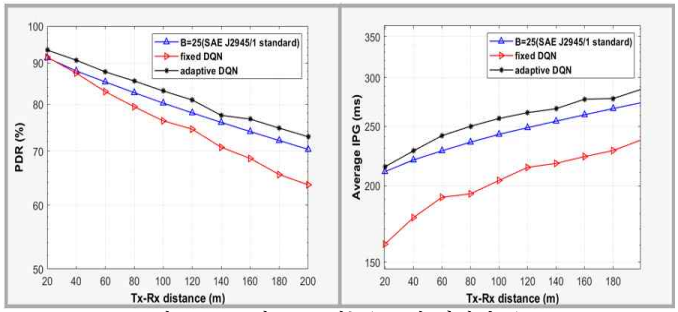

그림 2. PDR과 IPG 성능(600대 차량밀도)

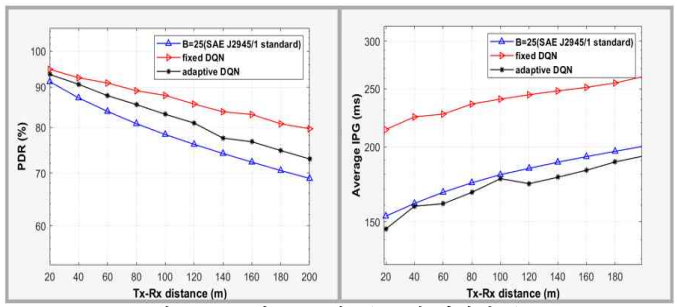

그림 3. PDR과 IPG 성능(400대 차량밀도)

ACKNOWLEDGMENT

본 연구는 국토교통부/국토교통과학기술진흥원의 지원(21AMDP-C160502-01)과 과학기술정보통신부 및 정보통신기획평가원의 Grand ICT 연구센터지원사업(IITP-2021-0-01462)의 연구결과로 수행되었음.

참 고 문 헌

[1] 정구선, 정창규, 문철, "C-V2X를 위한 지역 혼잡도 정보를 이용한 분산 혼잡 제어," 한국전자파학회논문지, vol. 32, no.2, pp.101-109, Feb. 2021.

[2] SAE International, on-board system requirements for V2V safety communications, Standard Doc. J2945/1, 2016.

[3] SAE International, on-board system requirements for LTE-V2V safety communications(under development), Standard Doc. J3161/1, Aug. 2019.

[4] Qualcomm, "R1-1611594. congestion control for V2V," 3GPP TSG RAN1-87, Tech. Rep., Nov. 2016.

[5] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529-533, Feb. 2015.

[6] Web page of LTEV2Vsim. [Online]. Available: https://github.com/alessandrobazzi/LTEV2Vsim