

레벨 4 자율주행을 위한 컴퓨팅/네트워킹 부하의 로드밸런싱 알고리즘 및 VTD-MATLAB 통합 시뮬레이터를 통한 분석

김정환, *이승재, **성원우, *조한신, **강원울, 곽정호, 최지웅
 대구경북과학기술원, *한양대학교, **㈜자동차공학연구소

ghks9876@dgist.ac.kr, *freitag97@hanyang.ac.kr, **wonwoo@ivh.co.kr, *hsjo23@hanyang.ac.kr,
 **wykang@ivh.co.kr, jeongho.kwak@dgist.ac.kr, jwchoi@dgist.ac.kr

Development of Load Balancing Algorithm for Computing/Networking Load for Level 4 Autonomous Driving and Analysis by VTD-MATLAB Integrated Simulator

Kim Jeonghwan, *Lee Seungjae, **Seong Wonwoo, *Jo Hanshin, **Kang Wonyul, Kwak Jeongho, Choi Jiwoong
 DGIST, *Hanyang University, **IVH

요약

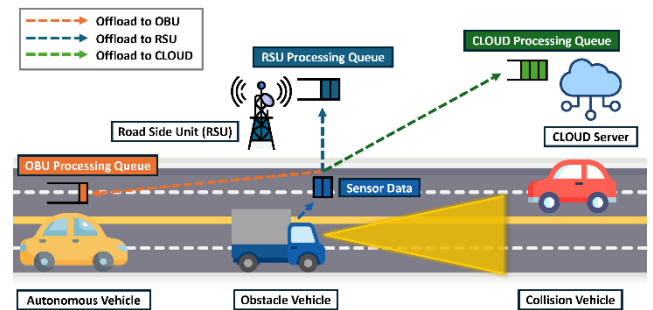
본 논문은 원활한 자율주행의 필수 요소인 객체 인식 기술의 복잡한 DNN 모델로부터 발생하는 많은 양의 연산을 효율적으로 처리하고자, 컴퓨팅/네트워킹 부하와 전력 소모량을 동시에 감소시키는 로드밸런싱 알고리즘을 제안한다. 또한 이를 VTD-MATLAB 통합 실시간 시뮬레이터에 구현하고, 성능을 분석함으로써 제안한 알고리즘이 기존의 알고리즘들보다 더 뛰어난임을 입증한다.

I. Introduction

최근 몇 년간 자율 주행에 대한 관심이 급증하면서 다양한 분야에서 연구가 진행되었다. 자율 주행의 핵심 중 하나는 이미지를 분석하여 객체의 위치를 파악하고 이를 식별하는 것인데, 오인식이 사고로 이어질 수 있는 잠재적 위험 때문에 높은 정확도를 요구하게 된다. 따라서 추론 정확도를 높이기 위해 점점 더 복잡한 Deep Neural Network (DNN) 모델이 개발되고 있지만 이는 차량 내부의 CPU 나 GPU 의 처리 시간을 필수적으로 증가시키게 된다. 현재는 이러한 연산을 차량 내부 처리 능력에만 의존하는데, 자율 주행에 필요한 데이터들을 모두 차량 내부에서만 처리하는 것은 전력 소비와 처리 속도 측면에서 매우 비효율적이며 원활한 자율주행에 문제를 야기할 가능성이 크다 [1]. 따라서 본 논문에서는 상기의 문제를 해결하기 위해 Lyapunov 최적화 기반 네트워킹-컴퓨팅 부하의 로드밸런싱 알고리즘을 제안하며, 이를 Vehicle Traffic Data (VTD) - MATLAB 통합 시뮬레이터를 통해 성능을 분석하고자 한다.

II. System Architecture

본 논문에서는 Fig 1 과 같은 차량 자체 On Board Unit (OBU)-Road Side Unit (RSU)-CLOUD 3 계층 시스템 아키텍처를 가정한다. 각 차량은 RSU, CLOUD 와 무선으로 통신하며 차량은 자신의 태스크를 차량 내부의 GPU, RSU, CLOUD 중 하나를 골라 처리하도록 할 수



있다. 여기서 알고리즘과 시뮬레이터의 확장성을 나타내고자 자율주행 차량 앞쪽에 장애물 차량으로 인해

Fig 1. System Architecture

원활한 센서 데이터 수집을 할 수 없는 상황을 가정한다. 장애물 차량이 자율주행에 필요한 센서데이터를 수집하면 이를 로드밸런싱 알고리즘에 따라 OBU/RSU/CLOUD 의 컴퓨팅 및 네트워킹 부하를 고려하여 도출된 장소로 오프로딩하여 처리하게 된다. 하나의 CLOUD 에 여러 RSU 와 OBU 가 연결되어 있으며, 각 RSU 또한 다수의 OBU 와 연결되어 있다. 차량은 가장 가까운 RSU 와 연결되고, 차량이 이동하면서 연결된 RSU 가 바뀌게 된다. 아키텍처의 각 구성요소는 별도의 처리 대기열 (bits)을 지니고 있다. 시스템의 대기열 수식은 다음과 같이 표현할 수 있다.

$$Q_i(t+1) = \left[Q_i(t) + (1 - \theta_j(t) - \sigma(t))a(t) - \frac{s_i(t)((1 - \theta_j(t) - \sigma(t)))}{\gamma} \right]^+, \quad (1)$$

$$Q_j(t+1) = \left[Q_j(t) + \theta_j(t)a(t) - \frac{s_j(t)}{\gamma} \right]^+, \quad (2)$$

$$Q_k(t+1) = \left[Q_k(t) + \sigma_k(t)a(t) - \frac{s_k(t)}{\gamma} \right]^+, \quad (3)$$

$Q_i(t), Q_j(t), Q_k(t)$ 는 각각 타임 슬롯 t 일 때 차량 OBU, RSU, CLOUD 의 처리 대기열을, $[x]^+ = \max(x, 0)$ 을 나타낸다. $\theta_j(t) \in \{0,1\}, \sigma_k(t) \in \{0,1\}$ 는 각각 RSU 와 CLOUD 로의 오프로딩 결정 변수이며, $a(t)$ 는 타임 슬롯 t 일 때 장애물 차량이 수집하는 센서 데이터의 크기이다. 센서 데이터는 단일 비트를 처리하는 데 필요한 사이클 수인 γ 를 처리 밀도로 가지고, $s_i(t), s_j(t), s_k(t)$ 는 각각 OBU/RSU/CLOUD 의 GPU 클럭 주파수를 나타낸다. 이 대기열 모델들을 활용하여 처리 지연 시간을 일정 수준 아래로 유지 및 차량의 시평균 전력 소모량을 최소화하는 문제를 고안하고, 이를 Lyapunov 최적화 기법으로 푸는 로드밸런싱 알고리즘을 제시한다. 그리고 이 알고리즘을 VTD-MATLAB 통합 시뮬레이터를 통해 구현하고자 한다.

III. VTD-MATLAB Simulator Results

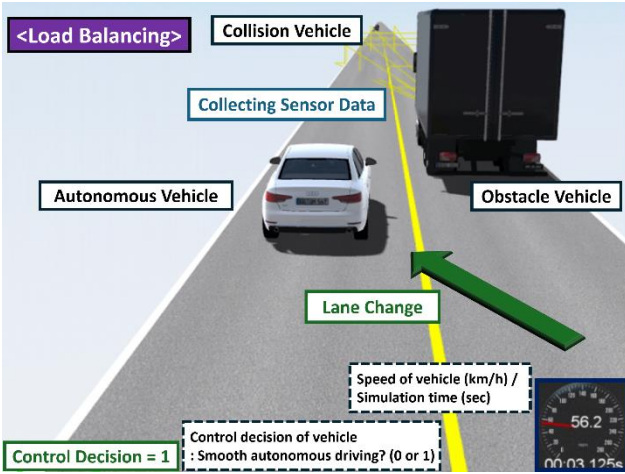


Fig 2. VTD Simulator Display

Fig 2 는 VTD-MATLAB 통합 시뮬레이터의 디스플레이 화면을 나타낸다. 매 타임 슬롯마다 MATLAB 의 알고리즘 결정 값을 받아 차량 데이터와 함께 화면에 표시되는 것이다. 화면에 표시된 건 제안한 로드밸런싱 알고리즘 시나리오이며 수집되는 센서데이터의 컴퓨팅/네트워킹 부하를 적절히 잘 처리하여, 시야가 없는 상황에서도 차선을 바꿔 추월할 준비를 하는 모습을 확인할 수 있다. Fig 3 은 비교 알고리즘과 제안한 알고리즘에서의 OBU/RSU/CLOUD 평균 대기열 길이이다. 비교 알고리즘은 다음과 같다. (1) OnlyOBU: 수집된 센서데이터를 모두 OBU 에서 처리한다. (2) OnlyOffloading: 수집된 센서데이터를 모두 RSU/CLOUD 에 오프로딩하여 처리한다.

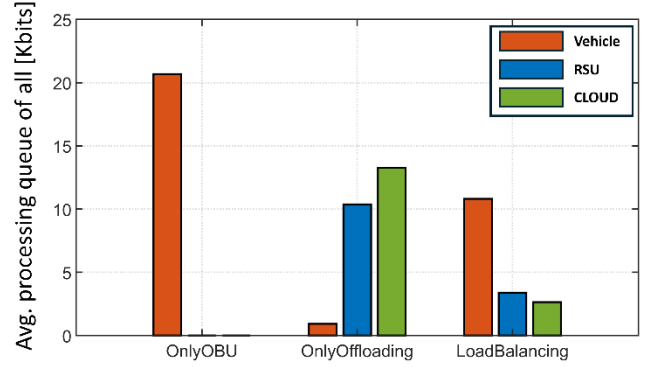


Fig 3. OBU/RSU/CLOUD 의 평균 대기열 길이

제안한 로드밸런싱 알고리즘이 다른 두 비교 알고리즘들에 비해 들어오는 센서 데이터를 어느 대기열에도 부하가 가지 않도록 잘 처리하는 것을 Fig 3 을 통해 확인할 수 있다. 각 대기열의 공정성을 고려함으로써 전체적인 처리 지연 또한 안정적으로 관리할 수 있다.

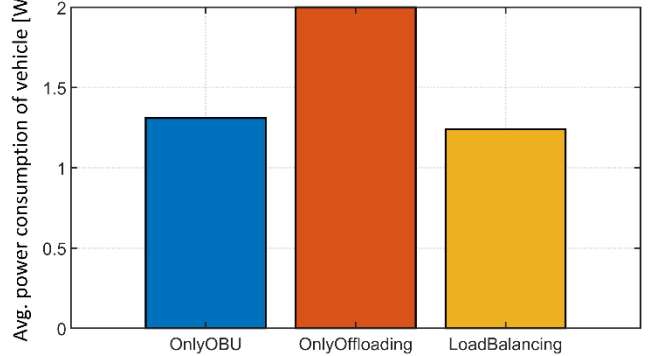


Fig 4. OBU/RSU/CLOUD 의 평균 전력 소모량

또한 Fig 4 을 통해서 처리 지연을 효과적으로 관리할 수 있음과 더불어 차량의 평균 전력 소모 또한 유의미하게 감소시킴을 확인할 수 있다.

ACKNOWLEDGMENT

이 논문은 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2022-0-01053, 다중 통신기술 네트워크 로드밸런싱 기술개발)

참 고 문 헌

- [1] L. Liu, S. Lu, R. Zhong, et al., "Computing systems for autonomous driving: State of the art and challenges," IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6469-6486, Apr. 2021.
- [2] M. Neely, "Stochastic network optimization with application to communication and queueing systems," Synthesis Lectures on Communication Networks, pp. 1-211, 2010.