

SMPL을 이용한 노변 네트워크 캐쉬 성능 분석기의 구현¹⁾

이정훈
제주대학교 전산통계학과
e-mail:jhlee@jeju.ac.kr

Implementation of a cache performance analyzer for roadside network based on SMPL

Junghoon Lee
Dept of Computer Science and Statistics
Jeju National University

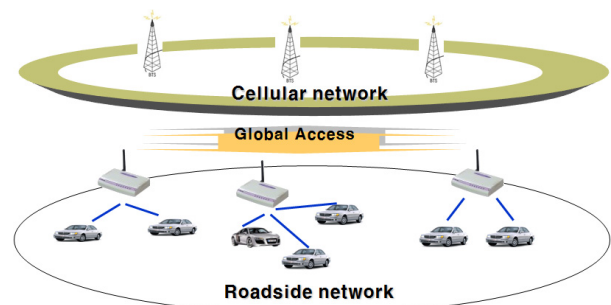
요 약

본 논문에서는 이산 이벤트 시뮬레이터인 SMPL을 이용하여 노변 네트워크에서의 데이터 처리에 따르는 데이터 캐쉬 성능분석기를 구현한다. 구현된 성능분석기는 SMPL의 요청 도착과 서비스 사건 처리를 기본 골격으로 하여 실제 차량의 궤적 데이터에 기반한 데이터 요청 생성부와 큐잉 정책과 캐쉬 정책을 선택할 수 있는 정책 처리부 등으로 구성된다. 이 분석기는 서비스율, 해당 정책, 캐쉬의 크기 등의 수행인자를 설정하여 이에 따르는 큐 길이의 분포, 캐쉬의 히트율, 요청 처리시간의 분포 등을 측정할 수 있도록 한다. 추정된 성능 요소를 기반으로 노변 네트워크에 기반한 차량 텔레매틱스 시스템에서 RSU(RoadSide Unit)의 배치, 성능 요구사항 분석, 새로운 큐잉 정책과 캐쉬 정책의 설계 등 다양한 응용이 가능하다.

1. 서론

차량형 텔레매틱스(vehicular telematics)는 차량에 내외부 유무선 통신 기능과 위치과악 장치를 설치하여 차량에 관련된 다양한 서비스를 광대한 지역에서 제공한다[1]. 차량 내의 텔레매틱스 장치는 일종의 내장형 컴퓨터(Embedded computer)로서 계산 능력을 갖고 있을 뿐 아니라 GPS 수신기와 같은 위치과악 장치, WLAN 혹은 CDMA(Code Division Multiple Access), 등과 같은 통신 인터페이스를 갖고 있어서 차량내 상태정보 수집과 제어기능은 물론 운전경로 추천, 차량 추적, 자동 사고정보 전파 등의 기능을 제공한다. 최근 차량형 텔레매틱스 네트워크 구성에 있어서 길가에 RSU(RoadSide Unit)를 설치하여 차량간 통신을 지원하는 노변 네트워크가 추진되고 있다[2].

노변 네트워크를 통해 RSU의 데이터가 중앙서버에 업로드되거나 반대로 서버의 데이터가 다운로드될 수 있다. 노변 네트워크에서 RSU로는 IEEE 802.11의 AP (Access Point)와 같은 장치가 사용될 수 있는데 (그림 1)에서 보는 바와 같이 차량과 RSU간에는 수십 Mbps의 대역폭이 지원되는 반면 이들이 글로벌 네트워크 연결시에는 셀룰라 네트워크와 같은 고비용 저대역폭 네트워크를 사용하게 되어 이 과정에서 병목현상이 발생할 가능성이 높다.



(그림 1) 노변 네트워크의 구조

따라서 주변의 교통상황 및 데이터 요청 패턴, 데이터 제공 방식 등 다양한 인자에 따르는 RSU의 성능을 파악하여 RSU의 배치 방식, RSU의 성능 요구사항, 효율적인 데이터 관리 정책의 개발 등에 대한 필요성이 대두된다[3]. 본 논문에서는 이러한 노변 네트워크에서의 RSU 서비스 성능 측정을 위하여 이산 이벤트 (discrete event) 시뮬레이터인 SMPL을 기반으로 하여 궤적 데이터 처리, 큐잉 동작을 위한 요청 및 서비스 처리, 요청 처리 큐를 위한 다양한 정책 라이브러리, 캐쉬 관리를 위해 필요한 정보 계산 함수들을 구현하였다[4].

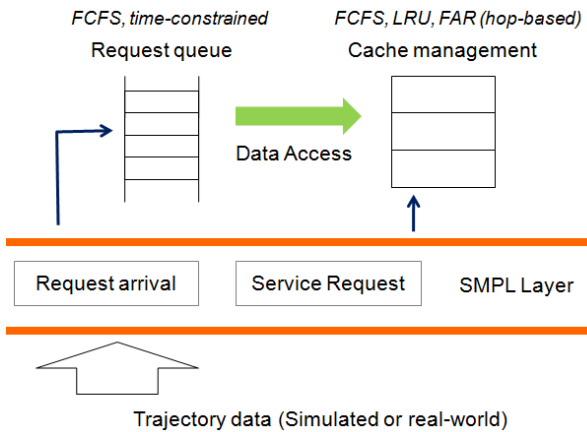
2. 성능분석기의 구현

본 논문에서 구현된 시뮬레이터의 구조는 (그림 2)에서 보는 바와 같이 SMPL 계층의 기본적인 큐잉 모델에 따라 요청의 도착과 큐잉 및 요청의 서비스 사건을 중심으

1) 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2009-C1090-0902-0040)

로 이에 대한 필요 모듈들을 구현한다.

SMPL 계층에 요청을 생성하여 전달하려면 우선적으로 요청이 도착하는 모델에 대한 정의가 필요하다. 가상적으로 생성된 모델은 포아송 분포나 waypoint 모델 등을 따를 수도 있으며 실제 차량의 이동 데이터나 요청 데이터가 가용한 경우는 이를 파일 혹은 데이터베이스에서 읽어서 schedule 함수를 통해 SMPL 이벤트로 등록한다. 본 논문에서는 제주 택시 텔레매틱스 시스템에서 수집된 데이터를 처리하도록 구성되었다[1]. 이 요청은 서비스율에 따라 하나씩 처리되며 서비스율의 설정은 요청의 도착시간과 주어진 부하에 의해 결정된다.



(그림 2) 성능분석기의 구조

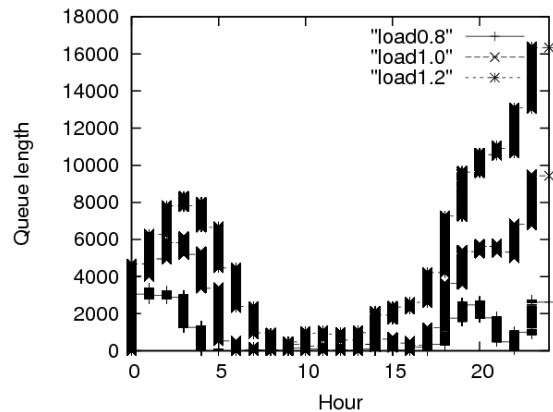
데이터에 대한 요청 패턴은 서비스와 차량의 이동 패턴에 따라 결정이 되는데 본 절에서 대상으로 하는 서비스는 고급 경로서비스로서 단순한 경로 결정이나 추천이 아닌 특정 경로에 대해 주변의 식당, 식당의 좌석가용성, 실시간 쿠폰, 교통상황 등 다양한 동적정보를 제공한다. 도로명 혹은 아이디어가 키가 되며 이를 기준으로 정보들이 검색된다. 또, 차량의 이동 패턴 파악을 위해 실제 궤적 데이터에서 출발지와 목적지를 추출할 수 있으며 이 두 지점간에 A* 알고리즘을 통해 차량이 이동하는 경로들의 모든 도로들을 추정하고 모든 도로들에 대한 정보가 검색된다고 가정한다. 본 시스템에서 24시간을 기준으로 총 94,546 개의 요청이 발생하였으며 이를 24시간에 처리할 수 있는 서비스 시간을 부하 1.0로 설정한다.

SMPL 계층에서 요청이 생성되면 이를 큐에 넣게 되는데 이 과정에서 데이터 조회 요청에 대한 스케줄링 정책이 구현 가능하다. 이 과정에서 요청 큐를 연결 리스트로 생성하였으며 요청 큐에는 FCFS, EDF(Earliest Deadline First) 등의 정책이 구현되어 선택이 가능하다. 또한 특정 각도나 공간적인 특성을 반영하는 스케줄링 정책의 구현과 결합이 용이하다. 데이터 조회 요청에 대한 서비스를 처리하는 과정에서 RSU의 캐쉬가 중요한 역할을 하는데 이는 데이터 저장의 용량적 한계와 시간적 한계를 고려하여 서비스 시간을 평가하는데 중요한 요소가 된다[5]. 평가 과정에서 캐쉬의 용량과 캐쉬 관리 정책을 설정할 수 있으며 캐쉬의 용량은 저장 가능한 데이터 아이템의 개수

를 단위로 가지며 관련된 정책은 FCFS, LRU,(Least Recently Used), FAR(Furthest Away First) 등이 포함되어 있다. 다양한 정책의 구현을 지원하기 위해 공간적인 관계를 결정하는 도로간의 거리계산, 특정 지점과 도로와의 각도계산 등의 함수들이 추가적으로 구현되어 새로운 캐쉬 관리 정책을 구현하고 평가할 수 있도록 하였다.

3. 수행 결과 및 결론

(그림 3)은 본 논문에서 구현된 성능분석기를 이용하여 요청 큐의 길이 변화를 측정한 결과이다. 이때 RSU의 위치는 차량의 통행량이 가장 많은 시내 중심지를 선택하였으며 이 위치에서 반경 무선 전송 거리에 해당하는 300m 내에서 출발한 탑승을 궤적 데이터로 사용하였다. 부하가 0.8, 1.0, 1.2의 경우에 대해 각 요청의 서비스 시간을 설정하였으며 부하가 높을수록 큐의 길이가 증가함을 알 수 있다. 특히 퇴근 및 저녁시간일수록 큐의 길이가 증가하여 효율적인 스케줄링 정책과 캐쉬정책의 효과가 큰 것을 예상할 수 있다.



(그림 3) 부하에 따르는 큐길이 분포

본 논문에서는 RSU에 기반한 차량 텔레매틱스 네트워크에서의 데이터조회 성능을 측정하기 위하여 SMPL을 기반으로 한 시뮬레이터를 구현하였다. 이를 기반으로 요청의 스케줄링 정책과 캐쉬 정책이 설계될 수 있어서 차량 네트워크에 대한 효율적인 계획이 가능할 것이다.

참고문헌

[1] J. Lee, G., Park, H., Kim, Y., Yang, P., Kim, P., and S. Kim, "A telematics service system based on the Linux cluster," LNCS, Vol. 4490, pp. 660-667, 2007.
 [2] Y. Zhang, J. Zhao, G. Cao, "On scheduling vehicle-roadside data access," ACM VANET, pp.0-18, 2007.
 [3] B. Yu, J., Gong, C., Xu, "Data aggregation and roadside unit placement for a vanet traffic information system," ACM VANET pp.49-57, 2008.
 [4] M. H. MacDougall, Simulating Computer Systems: Techniques and Tools, MIT Press, 1987.
 [5] Y. Ting, Y., Chang, "A novel cooperative caching scheme for wireless ad hoc networks: Group caching,," IEEE Int'l Conference in Networking, Architecture, and Storage. pp.62-68, 2007.