

차량 애드혹 네트워크에서 차량 자체 정보를 기반으로 한 클러스터링 기법

안 상 현[†] · 임 유 진^{**}

요 약

차량 애드혹 네트워크 환경에서 브로드캐스트 메시지 전송을 위한 효율적인 기법으로 클러스터링 기법이 있다. 대부분의 클러스터링 기법들은 안정적인 클러스터 구성을 위해 차량들 간에 정보를 교환하거나 이동성 정보를 계산하는 오버헤드를 야기한다. 이러한 오버헤드를 줄이기 위해 차량의 절대 속도를 기반으로 클러스터를 구축하는 CF-IVC[1]가 제안되었으나, CF-IVC의 경우 도로 혼잡 상황이나 운전자의 운전 행태를 고려하지 않음으로써 클러스터를 비효율적으로 구성하는 문제가 있다. 따라서 본 논문에서는 도로의 최고 제한 속도 및 도로 혼잡 상황을 고려한 차량 자체 정보 기반의 효율적인 클러스터 구축 기법을 제안한다. 제안 방식을 simple 플러딩 및 CF-IVC와 NS-2 시뮬레이션을 통해 비교함으로써 성능의 우수성을 입증한다.

키워드 : VANET(Vehicular Ad Hoc Network), Clustering, Broadcast

A Clustering Mechanism based on Vehicle Local Information in the Vehicular Ad Hoc Network

Sanghyun Ahn[†] · Yujin Lim^{**}

ABSTRACT

In the vehicular ad hoc network environment, the clustering mechanism is one of the efficient mechanisms to deliver broadcast messages. Most clustering mechanisms require message exchanges between vehicles to build stable clusters, which causes overhead. In order to reduce this overhead, CF-IVC [1] proposes the mechanism to construct clusters based on the vehicle speed. However, since CF-IVC does not consider the road traffic condition and the driver's behavior, it may result in inefficient clusters. Therefore, in this paper, we propose a mechanism to establish efficient clusters based on the vehicle local information with considering the road maximum speed limit and the road traffic condition. The performance of the proposed mechanism is validated by comparing with those of the simple flooding and CF-IVC through NS-2 simulations.

Keywords : VANET(Vehicular Ad Hoc Network), Clustering, Broadcast

1. 서 론

차량 애드혹 네트워크(VANET; Vehicular Ad Hoc Network) 환경에서 사고 알림 서비스 등을 위해서는 브로드캐스트가 필수적인 통신 기술이다. 브로드캐스트를 실현할 수 있는 가장 단순한 방법은 simple 플러딩을 사용하는 것으로, simple 플러딩은 broadcast storm 문제를 야기하며 이를 해결하기 위해서 효율적인 브로드캐스트 방식 중의 하나로 클러스터링 방식[1, 2]이 제안되었다.

클러스터링 방식에서 가장 중요한 이슈 중의 하나는 클러스터의 안정성을 유지하는 것으로, 클러스터 헤드로서 가장 적합한 노드를 클러스터 헤드로 정하기 위해서 노드들 간에 정보 교환을 하거나 노드의 이동성을 계산하게 된다. VANET의 경우 차량의 이동 속도가 고속이므로, 각 노드가 자신의 이웃 노드들에 대한 최신 정보를 유지하는 것이 쉽지 않다. 따라서 이러한 부정확한 정보들의 교환을 통하여 클러스터를 구성하게 되면 클러스터 구성의 비효율성과 불안정성으로 인해 클러스터의 재구성도 많아지고 오버헤드가 증가하게 된다. 보다 정확한 정보 유지를 위해서는 차량 간에 빈번하게 정보를 교환해야 하며 이 또한 오버헤드를 증가시킨다. 대부분의 클러스터링 방법들은 상대 속도(자신의 속도와 유사한 속도)를 기준으로 클러스터를 구성하며 이를

* 이 논문은 2010년도 서울시립대학교 교내학술연구비에 의하여 연구되었음.
† 정 회 원 : 서울시립대학교 컴퓨터과학부 정교수(교신저자)
** 정 회 원 : 수원대학교 정보미디어학과 조교수
논문접수: 2011년 9월 21일
심사완료: 2011년 10월 14일

위해 이웃 차량들과 속도 정보 등을 교환한다. 이와 같은 문제를 해결하기 위해 CF-IVC[1]는 절대 속도를 기준으로 차량들을 속도 그룹(speed group)으로 나눠서 클러스터 구성을 하며, 따라서 이웃 차량들과 속도 정보 등을 교환하지 않아도 된다. VANET의 경우 차량 속도에 제한이 있고 운전자의 운전 패턴이 있다는 특성을 고려해서 이런 로컬 정보들을 기반으로 클러스터를 구성한다면 클러스터 구성에 따른 오버헤드가 줄어들게 되며, 이로 인해 무선 링크의 충돌이 감소해서 데이터 전송 효율이 높아질 수 있다. 따라서 본 논문에서는 차량이 자신의 로컬 정보를 기반으로 클러스터를 구성하는 방법인 SLVC(Speed Limit based VANET Clustering) 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 제안하는 클러스터링 기법의 개념 및 동작 방식에 대해서 자세히 설명한다. 3절에서는 제안 방식의 성능을 기존의 단순 플러딩 방식 및 CF-IVC 방식과 시뮬레이션을 통해 비교 분석하며, 4절에서 결론을 맺는다.

2. 도로의 제한 속도를 고려한 VANET 클러스터링 방식

도로가 혼잡하지 않은 경우 일반적인 운전 패턴은 도로의 최고 제한 속도에 맞춰 운전하는 경향이 있으며, 최고 제한 속도를 기준으로 그 근처의 속도로 차량이 운행되는 경우가 대다수이다. 따라서 CF-IVC의 속도 그룹 개념으로 클러스터를 구성하면 CF-IVC 논문에서 언급한 것처럼 일부 몇 개의 (경우에 따라서는 하나의) 속도 그룹에 차량이 집중되는 현상이 생기게 되어 차량들을 속도 그룹으로 나누는 것의 장점이 없어지게 된다. 또한 속도 변화가 큰 차량의 경우도 평균 속도를 기준으로 속도 그룹이 결정되기 때문에 이런 차량을 클러스터 헤드로 결정하면 클러스터의 안정성이 떨어질 수 있다.

본 논문에서 제안하는 SLVC(Speed Limit based VANET Clustering) 기법에서는 각 차량이 주기적으로 자신이 클러스터 헤드가 될 수 있는지 여부를 어떤 기준에 의해서 결정하며, 이 기준은 각 차량이 자체적으로 정한 기준 속도 정보를 사용해서 결정된다. 기준 속도는 차량이 속한 클러스터의 CH로부터 수신한 CH_Announce 메시지에 포함된 CH의 속도를 기반으로 도로 혼잡 상황을 유추함으로써 결정되며, 초기 기준 속도 값은 도로 최고 제한 속도로 설정하고 그 후부터는 자신이 속한 클러스터의 CH 속도와 도로의 최고 제한 속도에 따라 정한다. 이렇게 각 차량이 기준 속도 정보를 유지하게 함으로써 기준 속도 근처에서 어느 정도 일정 속도로 운행하는 차량들만 클러스터 구성에 참여하게 하고, 그렇지 않은 차량(즉, 과속 또는 저속 차량, 속도 변경이 심한 차량 등)들은 클러스터 구성에 참여하지 않게 한다.

본 논문에서는 진출입구가 없는 고속도로 환경을 가정하며, 모든 차량은 도로 최고 제한 속도 및 다음에 정의될 파

라미터들의 값을 알고 있으며 동일한 값을 사용한다고 가정한다. 차량은 일직선 상에 배치된다고 가정하며 (즉, 도로에 굴곡이 없다고 가정하며), 데이터 전달 진행 방향은 한 방향(예, 후방)이라고 가정한다. 클러스터 크기는 클러스터 헤드를 기준으로 1-홉 반경으로 제한된다.

2.1 기준 속도 및 기준 속도 범위 파라미터 결정

클러스터 형성을 위해 각 차량이 유지하는 기준 속도, V_R 의 초기 값은 도로의 최고 제한 속도인 V_{max} 로 설정되며, 이후 V_R 은 클러스터 헤드 차량의 속도, V_{CH} 와 V_{max} 를 기반으로 조절된다.

$$V_{CH} = V_{max} \quad \text{if } V_{CH} > V_{max} \quad (1)$$

$$V_{CH,m} = V_{CH} + (V_{max} - V_{CH}) \times k_1$$

$$\text{where } 0 \leq k_1 \leq 1 \quad (2)$$

$$V_R = \omega_1 V_R + (1 - \omega_1) V_{CH,m}$$

$$\text{where } 0 \leq \omega_1 \leq 1 \quad (3)$$

$$V_R = V_{max} \quad \text{if } V_R > V_{max} \quad (4)$$

식 (1)은 V_{CH} 가 V_{max} 보다 큰 경우 V_{max} 로 함으로써 V_{CH} 가 V_{max} 보다 커지지 않게 하기 위한 것이다. 식 (2)는 클러스터 헤드의 현재 속도 V_{CH} 를 V_{max} 를 고려해서 $V_{CH,m}$ 으로 재계산한 것으로, V_{CH} 를 그대로 사용하지 않고 V_{max} 로부터의 차이를 반영함으로써 원래 도로의 최고 제한 속도 요소가 클러스터 헤드 선정에 반영될 수 있도록 하고 또한 도로 혼잡 상황 시 기준 속도가 너무 작아지는 것을 방지한다. 식 (3)은 가중 지수 평균(weighted exponential average) 방식으로 V_R 을 계산하는 것으로, 과거 기준 속도 이력을 반영함으로써 현재 클러스터 헤드의 속도에 의해서만 기준 속도가 정해지지 않도록 한다. V_R 의 초기 값은 V_{max} 로 설정한다. 식 (4)는 기준 속도가 V_{max} 보다 커지지 않도록 하기 위한 것이다.

V_R 이 크면 차량 밀도가 낮을 가능성이 높고 따라서 클러스터 내 차량 밀도가 낮을 가능성이 높으며 또한 차량 간 속도 차가 클 가능성이 높다. 제안 방식에서는 V_R 근처로 운행하는 차량들을 결정하기 위해 기준 속도 범위 파라미터 δ 를 사용하며, δ 는 현재 기준 속도를 기반으로 조절된다. V_R 이 큰 경우 차량 밀도가 낮을 가능성이 높기 때문에 δ 를 크게 하고, 반대로 V_R 이 작으면 δ 를 작게 조절한다. 다음 식 (5)는 기준 속도 범위 파라미터 δ 를 결정하는 수식이다.

$$\delta = V_R \times k_2 \quad \text{where } 0 \leq k_2 \leq 1 \quad (5)$$

2.2 클러스터 헤드 결정

기준 속도 근처로 운행하는 차량들의 집합을 CH 후보 그룹(CH candidate group; CH_{CG})이라 하며, CH 후보 그룹은 다음과 같이 정의된다.

$$CH_{CG} = \{N(i) \mid V_R - \delta \leq V_{N(i)} \leq V_R + \delta\}$$

where $V_{N(i)}$ 는 차량 $N(i)$ 의 속도 (6)

차량의 상태는 클러스터 헤드(cluster head; CH), 멤버(member; M), 게이트웨이(gateway; GW), 가상 클러스터 헤드(virtual cluster head; VCH)의 네 가지로 구분된다. 클러스터 헤드는 CH 후보 그룹에 속하는 차량 중에서 확률을 기반으로 선정된다. CH 후보 그룹에 속하는 차량 중 클러스터 헤드가 되지 못한 차량은 하나 이상의 클러스터에 멤버로 가입하며, 이때 두 개 이상의 클러스터에 멤버로 가입했거나 또는 이웃 차량이 다른 클러스터에 속해 있는 차량은 게이트웨이가 된다. CH 후보 그룹에 속하지 못한 차량은 클러스터에 가입하지 못하고 혼자만의 가상 클러스터를 구성하고 자신이 이 가상 클러스터의 VCH가 된다.

CH 후보 그룹에 속하는 차량 $N(i)$ 는 도로 최고 제한 속도 V_{max} , 기준 속도 V_R , 차량 $N(i)$ 의 평균 속도 $V_{avg_N(i)}$, 기준 속도 범위 파라미터 δ , 차량 $N(i)$ 의 속도 변화율 $V_{var_N(i)}$ 등의 로컬 정보를 기준으로 CH가 될 수 있는 확률인 $P_{CH_N(i)}$ 를 계산한다.

$$V_{avg_N(i)} = \omega_2 V_{avg_N(i)} + (1 - \omega_2) V_{N(i)}$$

where $0 \leq \omega_2 \leq 1$ (7)

$$P_{CH_N(i)_v} = |V_{avg_N(i)} - (V_R - \delta)| / ((V_R + \delta) - (V_R - \delta))$$

if $(V_R + \delta) \leq V_{max}$ (8)

$$P_{CH_N(i)_v} = |V_{avg_N(i)} - (V_R - \delta)| / (V_{max} - (V_R - \delta))$$

if $(V_R + \delta) > V_{max}$ (9)

$$P_{CH_N(i)_v_min} = P_{CH_N(i)_v} + v$$

where $0 < v < 1$ (10)

$$V_{var_N(i)} = \omega_3 V_{var_N(i)} + (1 - \omega_3) |V_{N(i)_p} - V_{N(i)}|$$

where $0 \leq \omega_3 \leq 1$ (11)

$$P_{CH_N(i)} = P_{CH_N(i)_v_min} + \alpha |V_{var_N(i)} - V_{var_max}| / V_{var_max}$$

where $0 < \alpha < 1$ (12)

$$P_{CH_N(i)} = 0 \quad \text{if } V_{var_N(i)} > V_{var_max} \quad (13)$$

$$P_{CH_N(i)} = 1 \quad \text{if } P_{CH_N(i)} > 1 \quad (14)$$

식 (7)은 차량의 평균 속도를 가중 지수 평균 방식으로 구하는 계산 식이며, $V_{avg_N(i)}$ 의 초기 값은 V_{max} 로 설정한다. 여기서 차량의 동적인 상황을 반영하기 위해서는 가중치 ω_2 를 작은 값으로 하는 것이 바람직하다. 식 (8)과 (9)는 차량의 평균 속도를 고려해서 클러스터 헤드가 될 수 있는 확률을 계산하는 것으로, 각각 CH 후보 그룹에 속하는 차량들의 속도 범위에서 최대 속도인 $(V_R + \delta)$ 가 V_{max} 보다 큰 경우와

작은 경우로 나뉘어서 계산한다. 이들 식은 V_{max} 에 가까울수록 클러스터 헤드가 될 수 있는 확률을 크게 하며, 이들 식으로 구해지는 CH 확률 값은 0과 1 사이의 값을 갖게 된다. CH 확률 값이 작을수록 다음에 정의될 τ 가 커져서 CH 결정을 위한 지연 시간이 증가하게 되며 반면 CH 수는 감소하게 된다. 따라서 본 연구에서는 CH 후보 그룹에 속하는 노드들이 갖게 되는 최소 CH 확률 값을 v 라는 파라미터 값으로 정의하며, 식 (10)은 v 를 고려한 CH 확률 계산식으로 CH 확률이 v 와 1 사이의 값을 갖도록 한다. 식 (11)은 차량의 속도 변화율 V_{var} 을 계산하는 식이며, $V_{N(i)_p}$ 는 이전 속도 감지 시점에서의 차량 $N(i)$ 의 속도이고 $V_{var_N(i)}$ 의 초기 값은 0으로 설정된다. 여기서 차량의 동적인 상황을 반영하기 위해서는 가중치 ω_3 을 작은 값으로 하는 것이 바람직하다. 식 (12)는 식 (11)에서 구한 차량의 속도 변화율을 고려한 CH 확률 계산식으로, V_{var_max} 는 속도 변화율의 상한 값이며 이 값을 초과하는 경우 해당 차량은 CH 후보 그룹에서 제외된다. 즉, 속도 변화율 V_{var_max} 이하인 차량에 대해서만 속도 변화율에 반비례해서 CH 확률을 최대 α 만큼 증가시킨다 (즉, $V_{var_N(i)}$ 가 0일 때 α 가 $P_{CH_N(i)_v_min}$ 에 더해진다). 식 (14)는 식 (11)에서 구한 $P_{CH_N(i)}$ 가 1보다 큰 경우 1로 재설정하기 위한 것이다.

차량 $N(i)$ 는 CH 확률 $P_{CH_N(i)}$ 를 기반으로 결정된 지연 시간 $T_{N(i)}$ 동안 CH-Announce 메시지를 다른 차량으로부터 받지 않는 경우 CH가 되고 CH-Announce 메시지를 1-홉 브로드캐스트하며, 클러스터 헤드 상태로 유지되는 동안에는 주기적으로 CH-Announce 메시지를 1-홉 브로드캐스트한다. 식 (15)는 $T_{N(i)}$ 를 결정하는 수식으로, 여기서 τ_{max} 는 차량이 CH-Announce 메시지를 브로드캐스트하기 전까지 기다려야 하는 최대 지연 시간이다.

$$T_{N(i)} = \tau_{max} |P_{CH_N(i)} - 1| \quad (15)$$

2.3 클러스터 형성

CH 후보 그룹에 속하는 차량이 CH 확률에 따른 지연 시간이 경과한 후 CH-Announce 메시지를 전송하기 전에 다른 차량으로부터 CH-Announce 메시지를 받으면 CH가 되는 것을 포기하고 CH-Announce 메시지를 보낸 CH를 자신이 속할 클러스터의 CH로 정한 후 자신의 상태를 M으로 설정한다.

상이한 CH-Announce 메시지와 VCH-Announce 메시지를 2개 이상 받은 차량이 GW가 된다. GW는 CH, VCH 간 데이터 전달을 중간에서 중계해주며, CH-Announce 메시지를 보낸 차량의 클러스터에 멤버로 가입한다.

CH 후보 그룹에 속하지 못한 차량은 VCH가 되며, VCH는 멤버 노드를 갖지 않는 클러스터 헤드로서 VCH-Announce 메시지를 1-홉 브로드캐스트한다. VCH-Announce 메시지를 수신한 차량은 해당 VCH-Announce 메시지를 더 이상 포워딩하지 않으며, 그 이유는 VCH-Announce 메시지의 목적이 단지 VCH에게 연결성만 제공

하기 위한 것이기 때문이다. 만일 VCH로부터 1-홉 내에 다른 차량이 존재하지 않으면 (즉, 1-홉 내에 VCH_Announce 메시지를 수신하는 1-홉 이웃 차량이 없으면) VCH는 단독으로 행동한다. 이렇게 하는 이유는, VCH가 되는 차량의 운행 패턴이 안정적이지 않기 때문에 이런 차량은 굳이 안정적인 클러스터에 멤버로 가입할 필요가 없기 때문이다. 오히려 VCH가 안정적인 클러스터에 멤버로 가입하면 해당 클러스터의 안정성을 약화시킬 수 있으며, 이로 인해 오버헤드가 증가하게 되는 문제를 야기할 수 있다. VCH의 역할은 다른 CH나 VCH에게 데이터를 포워딩하거나 이들로부터 데이터를 포워딩 받는 역할만 한다. 즉, VCH는 CH와 달리 멤버를 갖지 않기 때문에 멤버 관리 및 멤버로의 데이터 포워딩 기능은 수행하지 않는다.

두 개의 CH인 CH_A와 CH_B가 있을 때, CH_A가 CH_B로부터 CH_Announce 메시지를 받으면, CH_B로부터 CH_A까지의 홉수를 계산해서 CH_B의 클러스터 크기가 CH_A의 클러스터를 포함할 수 있을 정도로 크고 CH_A의 ID가 작으면 CH_A는 CH로서의 역할을 포기하고 CH_A의 클러스터 멤버들에게 CH_Merge 메시지를 보냄으로써 이들을 CH_B의 클러스터에 가입시킨다.

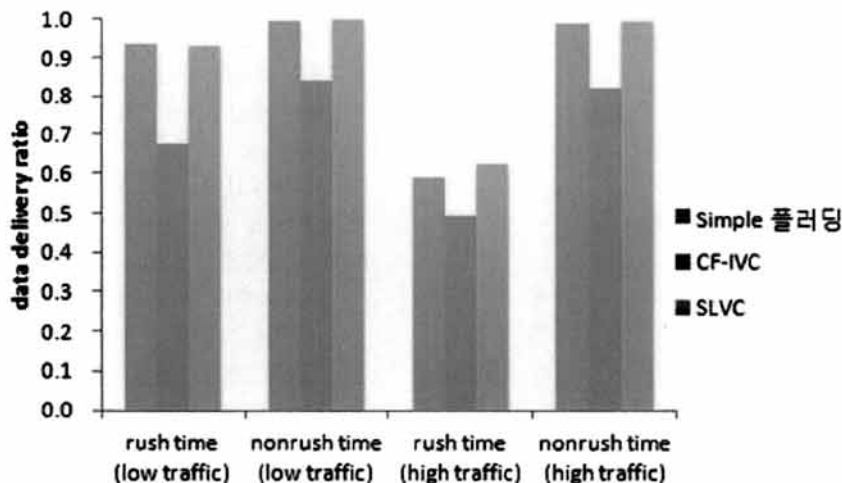
3. 성능 분석

본 논문에서 제안한 기법인 SLVC의 성능 분석을 위하여 NS-2[3] 시뮬레이터를 사용해서 simple 플러딩 및 CF-IVC 기법과 비교했으며, 성능 평가 요소로서 데이터 전송률(data delivery ratio)과 데이터 전송 지연시간을 사용했다. 데이터 전송률은 데이터 수신 대상이 되는 전체 차량 수에 대한 실제 데이터를 수신한 차량 수의 비율이며, 데이터 전송 지연시간은 소스 차량으로부터 데이터가 전송된 순간부터 전체 대상 수신 차량이 데이터를 수신할 때까지의 지연 시간이다.

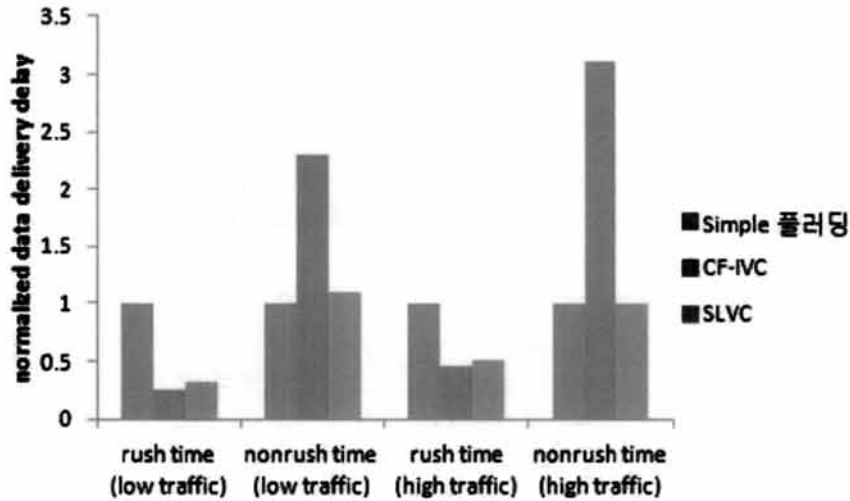
실험 환경은 편도 5개 차선으로 구성된 2km 범위의 고속도로로 설정했으며 신호 전파 모델은 자유공간 모델(free space model)을 사용했다. 각 차량의 전송 범위는 250m이며 MAC 프로토콜은 IEEE 802.11p[4]를 사용했다. 유효 데이터 전송률은 6Mbps이고 소스 차량은 도로 안전 메시지의 트래픽 생성률[5]을 기반으로 CBR(constant bit rate) 트래픽 생성기를 사용하여 초당 2 패킷을 생성하게 했다. 패킷 길이는 512byte이고 전체 실험 시간은 120초로 설정했다. 본 논문에서는 현실성 있는 고속도로 환경을 구현하기 위해 미국 캘리포니아 버클리 부근의 고속도로에서 차량의 속도 및 차량간 거리를 실측한 BHL[6] 실측 데이터를 사용했다. 본 실험에서는 시간을 RT(rush time)과 NRT(non-rush time)로 구분해서 차량의 평균 속도와 차량 간 거리를 다르게 설정했다. RT는 차량의 평균 속도와 차량간 거리를 25.5mile/hr와 13.7m로 설정하였으며, NRT는 차량의 평균 속도와 차량간 거리를 54.9mile/hr와 40m로 설정하였다[7]. 실험 환경 내 차량의 수도 RT와 NRT에 따라 각기 145대와 50대로 설정했다. 네트워크 내 백그라운드 트래픽(background traffic) 발생을 위해 유효 데이터 전송률의 20% 수준인 HT(high traffic) 환경과 유효 전송률의 10% 수준인 LT(low traffic) 환경으로 각기 나누어 실험했다[8]. <표 1>은 본 실험에서 사용한 제안 기법의 파라미터 값들이다. 파라미터 값들은 본 논문에는 포함되어 있지 않지만 예비 실험 결과를 기반으로 결정되었으며, 네트워크 환경에 맞는 적절한 파라미터 값들을 선정하는 것이 중요하다.

<표 1> 실험에 사용된 파라미터 값

파라미터	k ₁	k ₂	ω ₁	ω ₂	ω ₃	v	α	τ _{max}
값	0.5	0.4	0.01	0.01	0.01	0.4	0.4	5sec



(그림 1) 데이터 전송률



(그림 2) 정규화된 데이터 전송 지연시간

(그림 1)은 RT와 NRT일 때의 데이터 전송률을 측정 한 결과이다. 백그라운드 트래픽이 작은 경우(LT) SLVC 기법은 simple 플러딩 기법과 비슷한 수준의 높은 데이터 전송률을 보인다. 그러나 CF-IVC 기법은 차량의 속도를 기반으로 클러스터 형성 후 CH 선정 시 차량의 특성과 상관없이 가장 먼저 자신을 CH로 선언한 차량을 CH로 선정하기 때문에 CH가 갑자기 속도를 변경하는 경우 클러스터 내 CH가 부재하는 상황이 빈번하게 발생하게 된다. 따라서 LT의 경우에도 CF-IVC는 다른 두 가지 기법에 비해 76% 수준의 데이터 전송률을 보인다. 한편 네트워크 내의 트래픽 양이 많은 HT의 경우 RT일 때는 SLVC가 simple 플러딩 기법이나 CF-IVC에 비해 각기 6%와 26% 높은 전송률을 보였으며, NRT일 때는 simple 플러딩 기법과는 비슷한 성능을 CF-IVC보다는 18% 높은 성능을 보였다.

(그림 2)는 데이터 전송 지연시간을 측정 한 결과로 그림에서의 값은 각 실험 설정(RT에서의 HT/LT, NRT에서의 HT/LT)에서의 simple 플러딩 기법에서의 측정 값을 기준으로 정규화된 값이다. (그림 2)에서 보는 바와 같이 RT일 때는 SLVC 기법이 simple 플러딩 기법이나 CF-IVC 기법에 비해 LT인 경우 68%와 7%, HT인 경우 50%와 6%의 지연시간 감소를 보였다. CF-IVC 기법의 전송 지연시간이 상대적으로 우수해 보이는 이유는 CF-IVC 기법이 그림 1에서 보인 바와 같이 낮은 데이터 전송률로 인해 실제로 네트워크 내에 전송되는 메시지 수가 다른 두 기법에 비해 작기 때문이다. Simple 플러딩 기법의 낮은 성능은 CF-IVC와는 반대로 과도하게 많은 메시지가 네트워크 내에 전송되었기 때문이다. 결론적으로, SLVC는 simple 플러딩 기법 수준의 높은 데이터 전송률을 보이면서도 simple 플러딩 기법의 전송 지연시간의 50% 수준인 짧은 지연시간을 보임으로써 VANET 환경에 적합함을 알 수 있었다.

4. 결 론

본 논문에서는 각 차량이 도로의 최고 제한 속도, 클러스터 헤드의 속도, 자신의 속도 정보를 기반으로 도로 혼잡 상황을 유추하고 차량 간 정보 교환 없이도 자신의 클러스터 헤드 가능성을 결정해서 효율적으로 안정적인 클러스터를 구축하는 SLVC 기법을 제안했다. NS-2 시뮬레이션을 통해 SLVC가 simple 플러딩 기법이나 CF-IVC보다 데이터 전송률이나 데이터 전송 지연시간 측면에서 우수한 성능을 보임을 알 수 있었다.

참 고 문 헌

- [1] O. Kayis and T. Acarman, "Clustering Formation for Inter-Vehicle Communications," IEEE Intelligent Transportation Systems Conference, 2007.
- [2] C. Shea, B. Hassanabadi and S. Valaee, "Mobility-Based Clustering in VANETs Using Affinity Propagation," IEEE GLOBECOME, 2009.
- [3] The network simulator, ns-2, <http://www.isi.edu/nsnam/ns/>.
- [4] Q. Chen, D. Jiang, V. Taliwal, and L. Delgrossi, "IEEE 802.11 based Vehicular Communication Simulation Design for NS-2," ACM VANET, 2006.
- [5] G. Greneker, "Millimeter Wave Safety Warning System for In-Vehicle Signing", IEEE AES System Magazine, 1998.
- [6] The berkeley highway laboratory, <http://bhl.calccit.org:9006/bhl/>.
- [7] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in Sparse Vehicular Ad Hoc Wireless Networks," IEEE Journal on Selected Areas in Communications, Vol.25, No.8, pp.1538-1556, Oct., 2007.
- [8] S. Eichler, "Performance Evaluation of the IEEE 802.11p WAVE Communication Standard", IEEE VTC 2007.



안 상 현

e-mail : ahn@uos.ac.kr

1986년 서울대학교 컴퓨터공학과(학사)

1988년 서울대학교 컴퓨터공학과(석사)

1993년 University of Minnesota

컴퓨터학과(박사)

1988년~1989년 (주)데이콤연구원

1994년~1998년 세종대학교 컴퓨터학과 전임강사/조교수

1998년~현 재 서울시립대학교 컴퓨터과학부 정교수

관심분야: 에드혹 네트워크, 센서 네트워크, 홈 네트워크,
이동통신, 라우팅 프로토콜



임 유 진

e-mail : yujin@suwon.ac.kr

1995년 숙명여자대학교 전산학과(학사)

1997년 숙명여자대학교 전산학과(석사)

2000년 숙명여자대학교 전산학과(박사)

2000년 서울대학교 박사후연구원

2000년~2002년 UCLA 박사후연구원

2003년~2004년 삼성종합기술원 전문연구원

2004년~현 재 수원대학교 정보미디어학과 조교수

관심분야: 에드혹 네트워크, 센서 네트워크, VANET, 라우팅
프로토콜