VANET에서의 VMBC 알고리즘을 이용한 그룹 리더 선정 방법

김재명, 정은주, 정성대, 이상선
한양대학교 전자컴퓨터통신공학과

Selection Method of Group Leader based on VMBC Algorithm in VANET

Kim Jae-Myoung, Jung Eun-Ju, Jung Sung-Dae, Lee Sang-Sun
Hanyang University Department of Electronics Computer Engineering

Abstract - 서버와 클라이언트 기능을 모두 수행하면서 피어들끼리 직접 데이터를 주고받는 P2P 환경에서는 정보의 다양성으로 인하여 원하는 정보를 한 홉 이내에서 찾을 확률이 매우 낮다. 따라서 정보 검색을 위한 플러딩을 수행하게 되는데 플러딩 방법은 모든 피어가 네트워크에 참여하기 때문에 많은 부하가 생기고 네트워크 성능을 저하시킨다. 이러한 문제점은 차량 네트워크에서도 그대로 나타나게 된다.

이를 해결하기 위해 본 논문에서는 JXTA의 랑데부 피어 컨셉을 이용하여 정보 검색을 수행하도록 하였다. 랑데부 피어를 사용한 세 가지의 시나리오와 플러딩 방식을 사용하는 한 가지 시나리오를 시뮬레이션 한 결과 랑데부 피어를 사용하는 방식이 네트워크에 부하를 덜 주면서 효율적인 정보검색을 한다는 것을 확인하였다.

또한 차량 네트워크에서 이동성을 지원하기 위하여 기존의 클러스터링에 VMBC알고리즘을 이용한 그룹 리더(랑데부 피어)선정 방법을 제안하였다. 제안한 그룹 리더 선정 방법을 토대로 시뮬레이션 한 결과 이동성이 존재할 경우에도 플러딩 방식보다 효율적으로 정보 검색이 이루어지는 것을 확인할 수 있었다.

1. 서 론

급속한 인터넷의 발달로 정보의 홍수 속에 살고 있는 현대인에게 자동차는 정보를 제공받을 수 있는 또 하나의 생활공간으로 자리 잡고 있다. 또한 정보기술과 자동차 산업이 발전함에 따라 모바일 컴퓨팅 및 지능형 자동차에 대한 연구 개발이 활발히 이루어지고 있다. 이에 따라 텔레매틱스 서비스 제공을 위한 차량 간 효율적인 정보의 검색 방법에 대한 연구가 요구된다.

각 피어들은 서버와 클라이언트의 기능을 동시에 취하면서 피어들끼리 통신하는 방식을 통하여 데이터의 공유와 처리를 할 수 있게 한다[1]. 하지만 정보의 다양성으로 인해 P2P 네트워크에서 자신이 원하는 정보를 한 홉 이내에서 찾을 확률은 매우 낮다. 멀티 홉의 범위에서 정보를 검색하기 위하여 플러딩을 수행할 경우 모든 피어가 정보 전송에 참여하기 때문에 네트워크 부하로 인한 성능 저하를 초래하게 된다[2]. 이러한 문제점은 차량 네트워크 환경에서도 그대로 드러나게 된다. 따라서 피어들의 정보를 관리하는 그룹 리더 피어의 존재가 요구된다. 이러한 역할을 수행하기 위해 JXTA의 랑데부 피어 컨셉을 적용한다. JXTA란 현재 개발된 P2P 애플리케이션들이 각기 다른 시스템 환경으로 인해 상호간 통신을 할 수 없는 문제점을 해결하기 위해 개발된 기술이다[3].

기존의 랑데부 피어 컨셉은 랑데부 피어가 고정된 상태에서 피어들의 정보를 관리하였다. 따라서 이동성을 가지는 차량 환경에 이를 적용하기 위해서는 이동성을 지원하는 알고리즘이 추가적으로 필요하게 된다.

본 논문에서는 기존의 플러딩 방식과 비교하여 JXTA의 랑데부 피어 컨셉을 적용하였을 때의 정보 검색 효율성을 검증하고, 더 나아가 이동성이 존재하는 모바일 네트워크 환경에서 원활하게 검색을 수행하기 위한 랑데부 피어 선정 방안을 제시하고자 한다.

2. 본 론

2.1 JXTA의 랑데부 피어

Sun Microsystems사의 JXTA 기술은 네트워크 프로그래밍과 컴퓨팅 플랫폼으로 현대의 분산 컴퓨팅, 특히 P2P 컴퓨팅이나 P2P 네트워킹이라는 영역에서 폭넓게 발생하는 문제를 해결하기 위해 만들어진 것이다[4]. JXTA는 현존하거나 개발 중인 많은 P2P 시스템을 단순화할 수 있는 객체들을 가지고 있으며 다음과 같은 세 가지 특징이 있다.

첫째, 서로 연결된 피어들끼리 상대를 쉽게 찾아 서로 통신하며, 커뮤니티 기반의 활동에 참가하면서 서로 다른 P2P 시스템과 커뮤니티를 통해 중단 없이 서비스를 제공할 수 있도록 디자인된 상호 운용성(Interoperability)의 특징이 있다. 둘째, C나 자바와 같은 프로그래밍 언어, 윈도우나 유닉스 같은 시스템 플랫폼, TCP/IP나 블루투스와 같은 네트워킹 플랫폼에 독립적(Platform Independence)으로 설계되었다. 셋째, 센서나 가전제품, PDA, 네트워크 라우터, 데스크 탑 컴퓨터, 데이터 센터 서버 등 모든 디지털 장비에 적용할 목적으로 개발된 편재성(Ubiquity)의 특징을 갖는다.

2.2 MBC 알고리즘

본 장에서는 이동성 지원을 위한 알고리즘으로써 이동성 기반의 클러스터링 방법에 대한 논문에서 제안한 MBC(Mobility Based Clustering)알고리즘을 설명한다[5].

x와 y라는 노드가 L이하의 홉을 가지면 클러스터를 형성한다. 여기서 L은 네트워크 안정성에 의존하는 파라미터이다. 이 클러스터는 각 노드에 대해 상대적이기 때문에 C_i라는 집합으로 나타낸다. 이 집합을 기본으로 하여 첫 번째 단계에서는 이동성 정보에 대한 광고를 수행한다. 각 노드는 GPS로부터 자신의 위치정보를 얻게 되고 이 정보를 상용하여 자신의 속도를 계산할수 있다. 시간 t에서 노드 n의 속도는 수식(1)과 같이 표현한다.

$$v(n,t) \tag{1}$$

두 번째 단계에서는 이동성을 계산한다. 각 노드들은 첫 번째 단계를 거치면서 알게 된 주위 노드들과의 상대 속도를 계산하게 된다. 상대 속도 값을 기반으로 일정 시간(T) 동안 계산된 상대속도 크기의 평균을 구함으로써 상대적인 이동성을 계산하게 된다. 상대속도 및 상대적 이동성을 계산은 수식(2)(3)과 같다.

$$v(m,n,t) = v(m,t) - v(n,t) \tag{2}$$

$$M_{m,n,T} = \frac{1}{N}\sum_{i=1}^{N}|v(m,n,t_i)| \tag{3}$$

여기서 N은 상대속도를 계산한 횟수를 의미한다.

세 번째 단계에서는 임시적인 클러스터 헤드를 구성한다. 이동성 기준이 되는 Th 값 보다 작은 이동성을 가지는 노드 중 가장 낮은 ID를 가지는 노드가 임시적으로 클러스터 헤드로 선정된다. 수식(4)와 같다.

$$TCH = LST_{i \in s_m}\{ID | M_{m,i,T} < Th\} \tag{4}$$

네 번째 단계에서는 클러스터 병합을 수행한다. 임시 클러스터가 포함관계에 존재하면 자식 클러스터 헤드는 현재 클러스터 구성원들과 함께 부모 클러스터에 통합된다. 클러스터 헤드를 합병한 후에 최상위 클러스터 헤드는 새로 생성된 클러스터 헤드로 선택된다.

마지막으로 다섯 번째 단계에서는 클러스터 유지 및 재구축을 수행한다. 이동하려는 클러스터의 클러스터 헤드가 이동성 임계값 보다 낮은 이동성을 가진 경우에 한하여 클러스터를 이동할 수 있다.

2.3 VMBC 알고리즘

이동성이 존재하는 차량 환경에서는 고정적인 랑데부 피어를 사용하는 것에 대한 문제점이 발생한다. 따라서 본 장에서는 이동성이 존재하는 경우에도 안정적으로 정보 검색을 수행할 수 있도록 랑데부 피어를 설정하는 방법에 대해 논하고자 한다. 각각의 차량은 높은 속도로 도로 위를 이동한다. 하지만 같은 방향으로 이동하는 차량 간 상대속도는 비교적 높지 않다는 것을 알 수 있다. 이를 바탕으로 차량 간 통신을 수행하기 위한 환경 요인을 잘 갖추고 있다는 것을 전제로 하고, 랑데부 피어 설정 방안에 대해 연구하였다.

앞서 2.2장에서 설명한 이동성이 존재하는 모바일 Ad-hoc 네트워크상에서 상대적 이동성 값을 근거로 클러스터의 헤드를 선정하는 MBC 알고리즘은 차량 환경에서 랑데부 피어를 선정하고자 하는 목적에 잘 부합한다. 하지만 모바일 Ad-hoc 네트워크 같은 경우, 노드는 전 방향으로 이동할 수 있기 때문에 각 노드마다 주위 노드들과의 상대속도를 계산하고 이를 바탕으로 이동성을 계산하는 복잡한 절차를 거치게 된다. 일반 도로에서 차량은 일렬로 늘어서서 동일한 방향이나 역방향으로 이동하기 때문에 MBC 알고리즘 보다 간략화한 방법을 사용한 VMBC(Vehicle MBC) 알고리즘을 제안한다.

전송 범위 내에 존재한 노드들이 브로드캐스트(Broadcast) 방법을 사용하여 자신의 절대 속도를 주위 노드에게 알리게 된다. 1분을 기준으로 전파 범위(Radio Range)를 벗어나는 속도에 따라 등급을 분류하여 각 등급에 속한 노드들 사이에서 각 등급의 평균값과 가장 근접한 노드를 랑데부 피어로 선정한다. 선정된 랑데부 피어는 자신이 랑데부 피어임을 주위 노드에게 알린다.

랑데부 피어는 주기적으로 자신의 광고를 수행하는데 만약 랑데부 피어가 이동하면서 자신과 같은 등급의 속도를 가진 또 다른 랑데부 피어를 만났을 경우 두 랑데부 피어의 절대 속도를 비교한 후 평균값과 가장 근접한 노드가 랑데부 피어로 선정 되면서 두 그룹이 병합된다. 또한 동일한 등급의 노드 중에서 랑데부 피어의 브로드캐스트 범위에 존재하지 않아 광고를 수신하지 못한 경우에는 동일등급에 속하면서 자신과 인접한 타 노드에게 랑데부 피어의 정보를 요청하고 응답을 받은 후 랑데부 피어에게 광고를 등록하거나 질의를

하게 된다.

이러한 알고리즘을 사용하여 랑데부 피어를 선정함으로써 이동성을 가지는 차량네트워크에서 그룹리더를 적절히 유지할 수 있게 되고 이를 통해 기존의 플러딩 방식보다 더욱 높은 효율을 가져올 것이다.

2.4 시뮬레이션

정보를 검색하기 위해 랑데부 피어를 두는 방법과 랑데부 피어 없이 플러딩 방식을 사용하는 방법을 비교하기 위한 시뮬레이션을 하였다. 또한 앞서 제안한 VMBC 알고리즘을 적용하여 그 성능을 분석하였다. 랑데부 피어의 검색 효율을 보이기 위한 시뮬레이션에서의 실험영역은 170×70[m]로 설정하였고, 이동성이 존재할 경우 VMBC 알고리즘을 적용한 시뮬레이션의 실험 영역은 500×100[m]로 설정하였다. TwoRay 전파모델 및 전 방향 안테나를 사용하였다. TxPower는 802.11p 표준에서 가장 많은 채널이 사용하고 있는 23dbm을 사용하였다. 광고의 등록 및 정보 검색의 요청에 사용되는 제어 패킷의 크기는 30bytes, 데이터 패킷의 크기는 512bytes로 설정하였다. 시뮬레이션은 NS2를 이용하여 수행하였다[6].

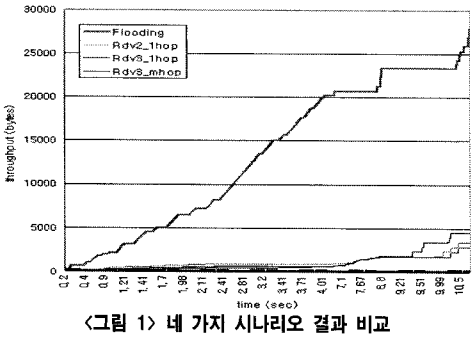
2.4.1 정보 검색 시나리오

앞서 제시한 랑데부 피어가 정보 검색을 하는데 있어서 검색 효율을 향상시킬 수 있는지 성능을 분석하기 위한 네가지 토폴로지를 설정한다. 한 홉 내에 랑데부 피어가 2개 존재할 경우(Scenario 1)와 3개 존재할 경우(Scenario 2)와 한 홉 이상의 거리에 랑데부 피어가 3개 존재할 경우(Scenario 3) 마지막으로 플러딩 방식을 사용할 경우(Scenario 4)로 나누어 성능을 분석하였다.

<표 1> 시나리오 별 패킷 전송량 비교 (단위 : byte)

	광고	정보검색	광고+정보검색
Scenario 1	0.9×10^3	0.7×10^3	1.6×10^3
Scenario 2	0.7×10^3	1.0×10^3	1.7×10^3
Scenario 3	0.7×10^3	1.2×10^3	1.9×10^3
Scenario 4	20.0×10^3	-	20.0×10^3

<표 1>에서 보듯이 플러딩 방식이 정보의 검색을 수행하지 않는 것을 감안 하더라도 적게는 10.5배, 많게는 12.5배 더 많은 제어 메시지가 전달되고 있다는 것을 알 수 있다. <그림 1>에서 이 네 가지 시나리오에 대한 결과를 그래프로 비교해보았다.

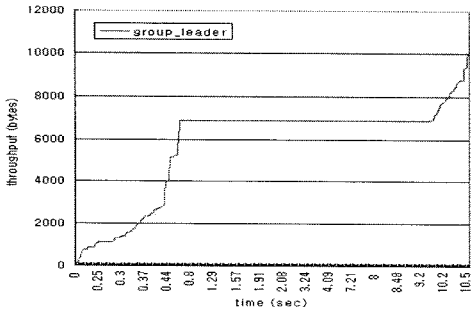


<그림 1> 네 가지 시나리오 결과 비교

이로써 네 가지 시나리오를 통해 플러딩 방식의 정보 검색보다 랑데부 피어가 존재할 경우의 정보 검색이 네트워크 성능 측면에서 더욱 효율적이라는 것이 입증되었다.

2.4.2 VMBC 알고리즘 적용 시나리오

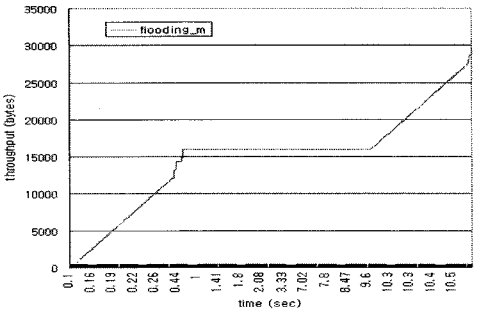
이동성이 존재하는 차량네트워크상에서 VMBC 알고리즘을 적용한여 그 성능을 분석하기 위한 다음과 같은 토폴로지를 설정한다. 10개의 피어가 이동하면서 처음으로 랑데부 피어를 설정하고 데이터 전송 절차가 모두 끝난 후에 10초가 되는 지점에서 새로운 랑데부 피어를 설정하도록 한다.



<그림 2> 이동성이 존재할 경우의 랑데부 피어 선정

0초부터 0.2초까지 자신의 속도를 주위 노드에게 브로드캐스트 하였고 0.2초부터 0.5초까지 광고 등록, 검색 질의, 데이터 전송의 절차를 거치도록 하였다. 10초부터 랑데부 피어를 재선정하는 절차에 들어간다. 10초부터 10.2초까지 랑데부 피어를 재선정하여 주위 노드들에게 알려주고 10.2초부터 10.5초까지 광고 등록, 검색 질의, 데이터 전송의 절차를 거친다. 초기 랑데부 피어 선정부터 이동성에 의해 토폴로지의 변화 후 랑데부 피어 선정까지 이루어지고 광고, 정보 검색, 데이터전송이 모두 완료되기까지 총 11×10^3bytes의 패킷이 사용되었다. 본 시나리오의 결과는 <그림 2>과 같다. 가로축은 시뮬레이션 시간으로써 단위는 sec이고 세로축은 전송 패킷량으로써 단위는 bytes이다.

VMBC 시뮬레이션과 동일한 환경에서 플러딩 방식을 사용할 경우와 비교하기 위한 시뮬레이션 결과는 다음 <그림 3>와 같다.



<그림 3> 이동성이 존재할 경우 플러딩 방식 사용

동일 조건에서의 시뮬레이션 결과, 플러딩 방식을 사용한 경우는 30×10^3bytes의 패킷이 사용되었다. 이는 VMBC알고리즘보다 약 2.7배 높다.

VMBC 알고리즘은 이동성이 없을 경우보다 오버헤드가 증가되지만 이동성이 존재할 경우의 플러딩 방식의 통신 보다 더욱 우수한 성능을 보여주었다.

3. 결 론

국토해양부 따르면 2007년 말까지 국내 자동차 등록대수가 1643만대로 나타났다. 우리나라는 세계 13위의 자동차 보유국으로써 인구 3명 당 1대의 자동차를 보유하고 있다는 통계 결과가 발표되었고 향후 3-5년 내에 인구 2.5명당 차동차 1대를 소유할 것으로 전망했다. 이처럼 자동차가 널리 보급되면서 도로에는 차량 인프라가 형성되게 되고 이를 활용한 서비스 제공에 대한 연구가 활발히 이루어지고 있다. 특히 이동성을 가지는 차량 네트워크에서 데이터의 안정적인 검색 및 공유 등의 서비스가 이루어지기 위한 분산 검색시스템에 대한 연구가 필수적 요소로 대두되고 있다.

효율적인 정보 검색을 위하여 JXTA에서 랑데부 피어를 사용하는 컨셉을 적용하였다. 또한 상대적 이동성 값을 근거로 클러스터의 헤더를 선정하는 MBC알고리즘을 차량 환경에서 랑데부 피어를 선정하기 위해 VMBC알고리즘을 제안하였다. 제안된 VMBC 알고리즘을 통해 정보의 검색 및 공유를 수행함으로써 이동성이 존재할 경우에도 토폴로지 변화에 따라 적절히 랑데부 피어가 선택, 유지되고 이를 바탕으로 안정적인 통신이 가능하게 된다. 따라서 기존의 플러딩 방식을 통한 정보 검색 방법보다 낮은 패킷 전송량을 가짐으로써 네트워크 부하를 줄여 효율이 향상된 점을 보이고 있다.

향후 임시적인 클러스터 구성을 위한 이동성 임계값 설정에 대한 연구 및 클러스터의 병합, 유지 및 재구축에 대해 다양한 상황에서의 구체적 알고리즘 연구가 수행되어야 할 것이다. 이는 추후 차량 간 네트워크에 대한 연구의 기반이 될 것으로 사료되며, 보다 효율적인 시스템이 구축될 것이라 예상된다.

ACKNOWLEDGEMENT

"본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-C1090-0801-0040)

[참 고 문 헌]

[1] Bisignano, M. Di Modica, G. Tomarchio, O. "JMobiPeer: a middleware for mobile peer-to-peer computing in MANETs", Distributed Computing Systems Workshops, 25th IEEE International Conference on 6-10 June 2005 pp. 785-791, 2005.

[2] 정왕부, 서현곤, 김기형, 손영호, "MANET 네트워크에서의 JXTA 기반의 P2P통신", 한국시뮬레이션학회 2005 춘계학술대회 논문집, pp. 139~143, 2005.

[3] C. Kim "A JXTA-based Architecture for Efficient and Adaptive Healthcare Services,", (Ed.): ICOIN 2005, LNCS 3391, pp. 776-785, 2005.

[4] "Industry Support for JXTA Technology Continues to Grow; Makes Significant Headway in First Year", Sun Microsystems Press Center 23 apr 2002.

[5] 안병구, "모바일 Ad-hoc 네트워크를 위한 이동성기반 클러스터링 방법", 한국인터넷방송통신TV학회 논문지, 제 5권 제 1호, pp. 11-17, 2005.

[6] 배성수, 한종수, "*Network Simulrator (NS2 기초와 활용)*", 세화, pp. 1-485, 2005.