

액티브 네트워크 기반의 고속 이동시 차량 간 통신 프로토콜*

장혜숙** · 이진관** · 정규철** · 이종찬** · 박기홍**

요 약

고속으로 이동 중인 차량들 간의 긴급한 정보를 주고받아 고속도로에서의 다중충돌사고를 미연에 방지할 수 있는 차량 통신 프로토콜을 제안한다. 본 논문에서는, 도로위의 차량들은 서로 연계성이 없기 때문에 위치추적시스템인 GPS에서 수신한 거리 값을 연계하여 클러스터를 구성한다. 클러스터를 IEEE 802.11 MAC 계층에서 구성하기 때문에 경로 설정으로 인한 과부하문제를 해소 하였다. 논문에서 제안한 액티브 네트워크 기반의 멀티 홉 라우팅 프로토콜은, Active Network를 이용하여 안정된 통신이 이루어 질수 있도록 한다.

The Network Protocol Among Cars at High-Speed based on Active Network

Jang Hae-Suk** · Lee Jin-Kwan** · Jung Kyu-Cheol**
Lee Jong-Chan** · Park Ki-Hong**

ABSTRACT

This paper propose CAR-TO-CAR protocol which can prevent the cars at high-speed from a multiple clash accident at highway with exchanging information. we construct cluster be linked with distance took from GPS(Global Positioning System) because there is no connection at cars on the road. The proposed method solved the overload problem by establishing route because construct the cluster at the IEEE 802.11 MAC class. 'Multi-hop Routing protocol' based on Active Network enable stable Network to be formed by using Active Network.

Key words : CAR TO CAR, cluster, Active Network

* 본 연구는 산업자원부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구 결과임.

** 군산대학교 컴퓨터 과학과

1. 서 론

오늘날은 고속화도로의 발달로 도로에서의 교통사고는 그 피해가 단일사고에 그치는 것이 아니라 후속 차량들의 어치구니없는 다중 추돌사고를 유발하고 있다. 이러한 교통사고로 인한 고속도로에서 발생하는 인명피해는 말할 것도 없고 경제적인 손실도 피해를 상상할 수 없을 정도로 크게 발생하고 있다. 그러하기 때문에 고속화 도로에서의 교통사고를 예방하기 위해서는 후속 차량에게 정보를 제공할 수 있는 방안이 모색되어야 한다.

전방의 차량에서 아무런 정보를 받지 못한 운전자는 앞차의 급브레이크 혹은 도로상의 장애물 등을 인지한 후 운전자가 브레이크를 작동하기까지 일반적으로 0.7~1.5초의 반응 지연시간이 소요되는 것으로 알려져 있다[1].

AASHTO(American Association of State Highway and Transportation Officials)의 조사에 의하면, 정보를 받고 반응하는 시간은 0.6초 정도였으며, 순수 인지에 의한 반응속도는 정보를 받고 반응하는 시간 보다 약 35% 정도 증가하는 결과를 보고하고 있다.

OFCOM(the Office of Communication) 조사에 따르면, 운전자가 교통위험 요소에 미리 대응 할 수 있는 1초의 여유를 갖게 되면 후면충돌을 90%까지 줄일 수 있다고 한다[2]. 이에 따라 전방차량의 위험상황 사실을 운전자에게 미리 제공한다면 교통의 안전성은 증대 될 것이다.

교통 안전통신의 적용범위는 일반 데이터 통신의 요구사항보다 훨씬 엄격하다. 핵심적인 요구사항으로는 낮은 지연시간, 전송신뢰도, 메시지 수신 비율 등이 있다[3]. 대부분의 교통안전관련 정보는 긴급 메시지의 형태로 전파되고, 이러한 메시지는 데이터 통신보다 훨씬 낮은 통신지연시간을 요구한다.

사용하려는 응용서비스에 따라 다르긴 하지만 일반적으로 1홉 간 발생하는 메시지 지연은 최대

100m/s이내로 요구된다[4]. 고속으로 이동 중인 차량 간의 특성상 시간적인 지연은 위험한 상황 대책을 위한 네트워크의 효율성을 매우 저하시킨다고 할 수 있다.

전송지연을 줄이기 위한 수단으로 단순한 브로드캐스트 방법을 통해 차량 간 1:1 전송을 수행한다면 패킷 충돌현상과 필요하지 않은 중복전송이 일어나 차량 안전통신의 요구사항을 만족 할 수 없게 된다. 운행 중인 차량 간에 정보 송수신이 직접 이루어지는 통신환경은 Ad-hoc 네트워크 모델과 일치한다. 일반적으로 차량 간 통신 기술은 Ad-hoc 네트워킹 기술에 바탕을 두고 있기 때문이다.

Ad-hoc 네트워킹 기술은 기지국이나 라우터와 같은 하부 구조 없이 이동 단말기 간에 무선통신을 가능하게 하는 기술이다. 이동 Ad-hoc 네트워크에서는 데이터 전송을 위해 유선과 달리 먼저 경로설정 알고리즘을 수행한 후 설정된 경로에 따라 데이터를 전송하는 방법이 대부분이다. 그러나 고속으로 이동 중인 차량들의 움직임으로 인해 경로유지가 어렵고 경로 손실에 따른 재 설정요구가 빈번하게 일어난다. 이 과정에서 이미 전송된 패킷은 손실이 되고 전체적인 패킷 수신율은 노드의 움직임 상태에 따라 달라 진다.

현재까지는 이러한 환경에서 데이터전송을 위한 알고리즘만 연구되었으며, 데이터의 신뢰성 향상을 위한 연구는 거의 이루어지지 않은 상태이다. 연계성이 없이 고속으로 이동 중인 차량 간에 거리를 계산하여 클러스터링을 하게 되면 설정된 거리 값으로 각 노드들의 연계성이 형성되고 차량사이의 거리 제한을 두기 때문에 무선 통신의 최대 도달거리가 250m이나, 200m의 거리로 제한을 두어 안정된 통신이 이루어지게 한다.

같은 구성원이 된 차량들 사이에서 거리 값으로 헤더를 선출하게 되고 선출된 헤더는 멤버 차량들에게 그룹 내 정보를 신속하게 전달하며, 고속으로 이동 중인 차량의 특성인 빈번한 토폴로지 변화로

인한 경로 이탈시 빠른 헤더의 재선출로 안정된 차량 간 통신을 가능하게 하였다.

2. 관련 연구

차량 간 통신에 적용될 수 있는 Ad-hoc 라우팅 프로토콜은 경로설정 여부와 데이터 포워딩 방식에 따라 토폴로지 기반 라우팅 프로토콜과 위치기반 라우팅 프로토콜로 분류 할 수 있다. 토폴로지 기반 라우팅 프로토콜은 주기적으로 또는 네트워크 토폴로지 상의 변화가 있을 때 마다 라우팅 정보를 네트워크 전체로 전파하여 모든 노드들이 항상 최신의 경로 정보를 유지하도록 하는 테이블 기반 관리 방식과 전송 트래픽이 발생하는 시점에서 경로를 탐색하고 경로 정보는 경로상의 각 노드에 임시적으로 저장되도록 하는 요구기반방식이 있다.

그리고 작은 범위의 네트워크에 대한 라우팅 정보는 테이블 기반 방식으로 유지하고, 작은 네트워크간의 패킷 교환이 발생할 경우 요구 기반 방식을 사용함으로써 두 방식의 장점을 결합한 하이브리드 라우팅 방식 등이 있다.

테이블 기반 방식의 라우팅 프로토콜은 패킷 발생시, 지연시간 없이 항상 최적의 경로를 통해서 패킷을 전송 할 수 있다는 장점을 가지고 있는 반면에, 전체 네트워크의 토폴로지 정보를 항상 유지해야 하기 때문에, 네트워크 토폴로지의 변화가 빈번한 차량 간 통신환경에서는 라우팅 부하로 인한 성능 저하를 초래 할 수 있다. 요구기반 방식의 라우팅 프로토콜은, 테이블 유지를 위한 라우팅 부하를 감소시키는 장점이 있는 반면, 트래픽이 발생했을때 경로를 설정하기 때문에 긴급 상황을 요하는 차량 간 통신에서의 신속한 정보전달의 지연시간을 초래한다.

DSR, AODV와 같은 요구기반 방식의 라우팅 프로토콜은 필요에 따라 RREQ(Route Request) 패킷을 네트워크 전체에 flooding하여 목적지까지의 경

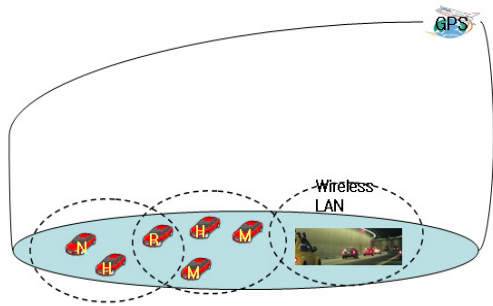
로를 찾는다. DSR에서는 RREQ 패킷에 지나온 노드마다의 ID를 패킷에 기록했다가 RREP (Route Reply) 패킷을 통해 송신지 노드에게 알려준다.

AODV에서는 RREQ 패킷이 지나온 노드마다 reverse-path에 대한 정보를 라우팅 테이블에 추가하고, RREP 패킷이 reverse-path를 통해 송신자에게 전달될 때, 송신자에서 목적지 사이의 경로가 중간 노드들의 라우팅 테이블에 추가된다. 이와 같은 DSDV, DSR, AODV는 모두 네트워크 토폴로지에 토대를 둔 라우팅 프로토콜로서, 규모가 큰 Ad-hoc 네트워크에는 적합하지 않다. 먼저, DSDV의 경우 네트워크의 규모가 커지면 주고받는 라우팅 정보의 양이 커지게 되고, 노드마다 유지하는 라우팅 테이블의 크기가 커지는 문제가 있다. 그리고 DSR과 AODV의 경우, 경로를 찾기 위해 RREQ 패킷을 전체 네트워크에 flooding해야 하므로 네트워크의 규모가 커질수록 경로설정을 위한 오버헤드도 커지게 된다.

또한 DSR의 경우, 네트워크의 규모가 커지면 소스 라우팅을 위한 ID의 리스트 길이가 늘어나기 때문에 RREQ 패킷과 RREP 패킷, 그리고 경로 캐시(Route cache)의 크기가 지나치게 커질 수 있다. 매초마다 변하는 차량들의 위상은, 이와 같은 일만 Ad-hoc 네트워크의 프로토콜 방식으로는 신속하고, 정확한 패킷 전송을 기대 할 수가 없다. Ad-hoc 네트워크의 프로토콜 방식의 하나인 AODV 방식에서는 위의 설명과 같이 네트워크내의 각 노드들이 활성 경로에 대한 정보만 유지하고 예비 경로에 대한 정보는 유지 않기 때문에 경로 단절시 마다 경로 재설정 과정을 수행해야 하기 때문에 경로 재설정시 시간을 필요로 한다.

본 논문에서는 차량 간 통신 환경에서 빈번한 네트워크 토폴로지의 변화 및 전송 실패로 인한 제어 패킷부하의 감소를 위한 새로운 멀티 홉 라우팅 프로토콜인 액티브 네트워크기반의 차량 간 멀티 홉 라우팅(Active Network Vehicle to vehicle Multi-hop Routing)을 제안한다. 액티브 네트워크

기반의 차량 간 멀티홉 라우팅 프로토콜은 IEEE 802.11 MAC 계층을 확장하여, 최선 경로를 설정하기 위한 라우팅 부하로 인한 성능을 저하시키는 문제를 해결 하였다. 차량 간 통신의 특성상 패킷 전송은 신속한 처리를 요구하므로 평소에 최선 경로를 확보 해두는 것은 중요한 문제이다. 또한 최선 경로를 확보하기 위하여 트래픽 과부하로 인한 성능 저하는 차량 간 통신에서 또 다른 문제를 야기 시킨다. 액티브 네트워크기반의 차량 간 멀티홉 라우팅에서는 최선경로 유지를 트래픽과 관련 없는 MAC계층에서 그룹으로 관리하고, 긴급 상황이 발생하여 패킷을 전송 할 때 에는 MAC계층에서 형성된 그룹단위의 멀티캐스트 전송방식으로 데이터를 포워딩하기 때문에 이러한 문제가 모두 해결 된다.



(그림 1) 이동 차량 간 멀티 홉 거리추정 클러스터링 구조

(그림 1)은 GPS로부터 위치 값을 수신하여 클러스터를 구성하는 그림을 개략적으로 나타내었다. (그림 1)에서와 같이 본 논문에서는 GPS를 통해 위치 값을 얻게 되고 위치 값을 기반으로 클러스터를 형성하게 된다. 클러스터 헤더의 역할은 클러스터 내의 멤버들에게 긴급한 상황 발생시 방송자 역할을 하게 된다. 클러스터 헤더가 긴급 상황을 방송하면 모든 클러스터 멤버들이 동시에 수신할 수 있게 된다.

3. 액티브 네트워크기반 차량 간 멀티홉 라우팅 프로토콜

이 장에서는 본 논문에서 제안하고 있는 액티브 네트워크 기반의 고속이동시 차량 간 통신 프로토콜을 기술한다.

본 논문에서 제안하는 액티브 네트워크 기반의 고속이동시 차량간 통신 프로토콜은 평소에는 Mac 계층에서 거리 값을 포함한 비콘 프레임은 주기적으로 전송하여 차량 간의 관계를 형성하고, 이동 차량들을 Group으로 묶어 관리한다. Mac계층에서 주기적인 비콘 프레임 전송은 경로를 설정하기 위한 라우팅 오버헤드와는 관계가 없기 때문에 이동 차량 간 통신에서의 빈번한 토폴로지 변화로 인한 라우팅 오버헤드는 발생하지 않는다. 긴급 통신요구 상황이 발생하면 멀티캐스트 방식으로 그룹의 헤더에게 긴급 상황이 전달되고 그룹헤더는 그룹 멤버들에게 메시지를 전달하게 된다. 그룹헤더와 Relay Car의 사이에는 액티브 네트워크가 설정되어 헤더의 이탈을 감시, 헤더의 이탈시 에는 헤더를 신속하게 재선출, 고속 이동시 전방차량의 긴급 메시지 전송을 후방차량들에게 신속하게 함으로써 치명적인 사고를 미연에 방지 할 수 있도록 하였다.

3.1 클러스터 구성

이동 차량들을 어떠한 기준에 따라 묶은 그룹을 클러스터라 하고 이렇게 이동 차량들을 그룹화 하는 것을 클러스터구성이라 한다. 이때 클러스터 내에서 차량들 간의 최대 경로 길이에 따라 1홉 클러스터, 2홉 클러스터, 3홉 클러스터 등으로 구분된다. 클러스터는 서버적인 클러스터 헤더와 클러스터 헤더의 서비스를 받는 클러스터 멤버차량, 그리고 2개 이상의 클러스터 헤더로부터 서비스를 받는 인접차량으로 구성된다.

제안된 방법에서는 정보전달 구조를 생성하기

<표 1> 확장된 비콘 프레임과 이웃노드 테이블 구조

Frame Control	Duration	DA	SA	BSSID	FCS	x_ axis	y_ axis	z_ axis
---------------	----------	----	----	-------	-----	---------	---------	---------

위해서 차량의 위치정보를 이용하기 때문에, GPS (Global Positioning System)와 같은 이동 차량에 설치 가능한 비교적 정확한 측위 시스템이 포함되어 있다고 가정한다. 모든 차량은 GPS를 통해서 차량의 위치와 속도를 수신하고, 차량 간 통신을 한다. 또한 하부의 전송계층은 고속의 이동성을 지원하는 무선 전송기술을 이용하여 무선랜과 유사한 통신반경을 제공하고 한 홉간 브로드캐스트는 CSMA(Carrier Sense Multiple Access)와 같은 기법을 통하여 무선 범위 안에서 충돌이나 간섭이 없이 모든 차량에게 무사히 전달됨을 가정한다. 차량정보를 수신한 차량은 상대거리를 계산한다. 차량 간 상대거리는 주변 차량정보 가운데 클러스터를 형성하는 판단기준이 되는 정보가 된다. 차량 간 긴급메시지 전송을 위해 사용할 무선 네트워크는 IEEE 802.11a 무선랜 기술을 기반으로 설계 되었다[5]. 무선랜에서 도달거리는 실내에서는 30m, 실외에서는 150~250m 정도를 예상하고 있으나, 주변 환경의 영향을 받아 실제 거리는 많은 차이를 보인다.

본 논문에서 제안하는 알고리즘은 무선랜 도달 거리를 고려하여 Relay Car에서의 200m지점의 차량들을 클러스터링 하고 Relay Car로부터 100m지점의 가장 가까운 차량을 클러스터 헤더로 설정 한다. Relay Car에서 무선랜 도달 거리보다 작은 200m 지점의 차량들을 클러스터링 하는 이유는 메시지 전달의 신뢰성을 높이고, 수신감도의 끊김 현상을 줄이기 위해서이다.

Relay Car를 각 차량들이 네트워크에 참여하는 이동차량들과 자동적으로 클러스터를 유지하기 위해서는 기본적으로 다음과 같은 행동 정보가 필요하다. 앞으로 차량 간 통신환경에서는 GPS기반 위치정보를 이용한 라우팅이 사용된다고 가정하

면, 이러한 라우팅을 위해서 모든 노드들은 주기적으로 자신의 정보가 포함된 비콘 을 전파하여 주변 노드들이 자신의 위치를 저장하도록 한다[6]. MAC 계층이 이처럼 확장된 비콘을 전파하고 그 정보를 수집하게 되면 각 노드들의 위치 정보 및 연결성 정보를 갖게 된다.

주기적인 위치정보 수집은 주기적인 비콘을 통해 각 노드는 MAC 계층에서 <표 1>과 같은 이웃노드 테이블을 구성할 수 있다[6]. 이 테이블은 전송범위내의 노드들의 정보만 기록되므로 그 크기 및 검색시간에 대한 오버헤드는 크지 않다.

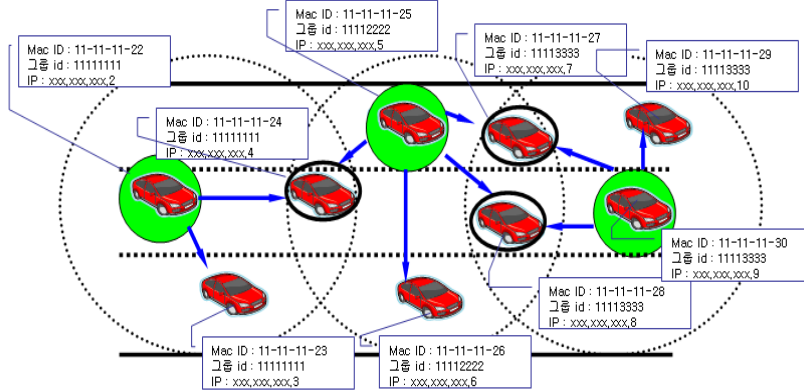
고속도로에 진입한 차량은 <표 1>의 비콘 프레임 테이블에서 획득한 이웃차량의 거리 값으로 클러스터를 구성하며 알고리즘은 다음과 같다.

```

PROCEDURE cluster-complet ;
begin
    send beacon in broadcast radius
    receive in-car in beacon radius
    if in-car = ack then
        if rp-location = true then
            common member broadcast
            insert cluster-id
            if relay-car = ack then
                accept relay-car
                select cluster header
            end
        end
    else
        create cluster-id
        if relay-car = ack then
            accept relay-car
            select cluster header
        else
            insert relay-id
        end
    end
end ;
    
```

Header Table			Relay car Table		
Mac id	IP	그룹 id	Mac id	IP	그룹 id
11-11-30	xxx.xxx.x.9	111	11-11-27	xxx.xxx.x.xxx.	11111
11-11-30	xxx.xxx.x.9	133	11-11-27	xxx.xxx.x.xxx.	3333
11-11-30	xxx.xxx.x.9	33	11-28	xxx.xxx.x.xxx.	1111
			11-28	xxx.xxx.x.xxx.	3333

Member Table			Relay car Table		
Mac id	IP	그룹 id	Mac id	IP	그룹 id
11-11-30	xxx.xxx.x.9	111	11-11-27	xxx.xxx.x.xxx.	11111
11-11-30	xxx.xxx.x.9	133	11-11-27	xxx.xxx.x.xxx.	3333
11-11-30	xxx.xxx.x.9	33	11-28	xxx.xxx.x.xxx.	1111
11-28	xxx.xxx.x.10	133	11-28	xxx.xxx.x.xxx.	3333
11-28	xxx.xxx.x.10	33			



(그림 2) 인접 차량 생성과 table 구성

차량이 고속도로에 진입 후 다른 차량들과의 거리 값을 알기 위하여 PS수신기로부터 수신한 거리 값을 포함한 비콘 프레임을 브로드캐스트 한다. 응답 비콘 프레임 수신 확인을 위해 설정된 거리 값 200m 이내에 차량이 있는지 여부를 확인 한다. 만약 200m 이내에 차량이 없을 때에는 자신이 클러스터 헤더가 되어 클러스터를 생성한다. 클러스터를 생성한다는 의미는 그룹 ID를 생성한다는 의미이다. 수신된 클러스터 헤더 개수를 확인하기 위해 인접차량을 선출하기 위해서 이웃 클러스터 헤더로부터 메시지 전송 여부를 확인한다. 이웃 클러스터 헤더로부터 메시지가 수신 되었으면 인접 차량으로 설정한다. 이웃 클러스터 헤더로부터 메시지가 수신되지 않은 경우 현재 클러스터 헤더의 멤버가 되어 그룹 ID를 부여 받는다. (그림 2)는 인접 차량이 생성하는 테이블과 클러스터 헤더가 생성하는 테이블을 나타내고 있다. 1개의 클러스터

헤더에서만 응답을 받은 경우 응답을 받은 차량은 송신 차량의 일반 멤버차량이 된다. (그림 2)에서와 같이 2개 이상의 클러스터 헤더로부터 응답을 받으면, 응답을 받은 차량은 인접차량이 되고, 클러스터 헤더 선출권이 주어지며, 인접차량 table이 구성되어진다. 다음은 클러스터 완성 알고리즘이다.

```

PROCEDURE cluster ;
begin
    send beacon in cluster radius
    receive in-car in beacon radius
    if in-car = false then begin
        if in-car > 1 then
            accept relay-car
        else
            create cluster-id
            insert cluster-id
        end
    else
        create cluster-id
    end
end ;
    
```

차량이 고속도로에 진입 후 다른 차량들과의 거리 값을 알기 위하여 GPS수신기로부터 수신한 거리 값을 포함한 비콘 프레임을 브로드캐스트 한다. 응답 비콘 프레임 수신 확인을 위해 설정된 거리 값 200m 이내에 차량이 있는지 여부를 확인한다. 만약 200m 이내에 차량이 없을 때에는 자신이 클러스터 헤더가 되어 클러스터를 생성한다. 클러스터를 생성한다는 의미는 그룹ID를 생성한다는 의미이다. 수신된 비콘 프레임의 거리 값이 200m 이내인가? 확인 후 수신된 비콘 프레임의 거리 값이 200m인 경우 같은 멤버 차량이 되어 같은 그룹 ID를 갖게 된다. 인접 차량을 생성하기 위하여 이웃 클러스터 헤더로부터 메시지가 수신되었는지 여부를 확인한다. 이웃 클러스터 헤더로부터 메시지가 수신되지 않은 경우 현재 클러스터 헤더의 멤버가 되어 그룹 ID를 부여 받는다. 이웃 클러스터 헤더로부터 비콘 프레임을 수신한 경우 인접차량의 테이블에 ID가 등록되며, 인접차량이 생성된다. 수신된 비콘 프레임의 거리 값이 200m 이상일 경우 응답 하지 않는다.

3.2 액티브 네트워크 차량 간 통신을 위한 멀티홉 라우팅 프로토콜 패킷 데이터 전송

액티브 네트워크 차량 간 통신을 위한 멀티홉 라우팅 프로토콜 패킷 데이터 전송은 이동 Ad-hoc 네트워크의 프로토콜인 요구기반 방식의 프로토콜에 의한 통신한다. 클러스터 헤더의 존재를 확인한 후 클러스터 헤더의 부재를 인식하게 되면 Active 패킷이 지시하는 내용을 노드의 순간적 실행환경에서 실시간으로 실행한다. 그리고 노드자신을 Active Node로 변경하여 경로 재설정 패킷을 송신하면서 새로운 경로를 찾게 된다. 다음은 액티브 네트워크 차량 간 통신을 위한 멀티홉 라우팅 프로토콜 패킷 데이터 전송을 위한 알고리즘이다.

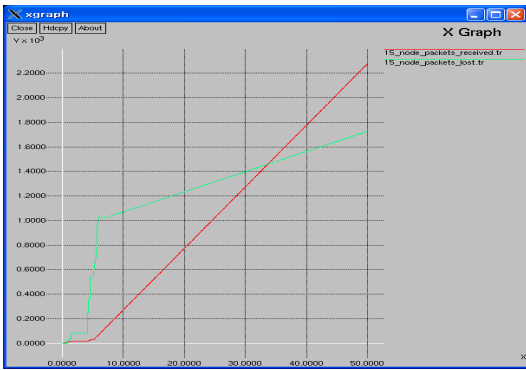
```

PROCEDURE avrm-packet-send ;
begin
    send RREQ in broadcast radius
    receive in-car in RREP radius
    if in-car = false then
        send RERR in broadcast radius
        begin
            else
                transmute AODV to AVRMM
                send AVRMM broadcast
                execute AVRMM active-code
                if cluster-hd = false then
                    transmute data-packet
                    to active-packet.
                    send active-node
                    in broadcast radius.
                    send data-packet interrupt
                    begin
                        else
                            send RREQ in broadcast radius
                        end
                    end
                end
            end ;
    end ;

```

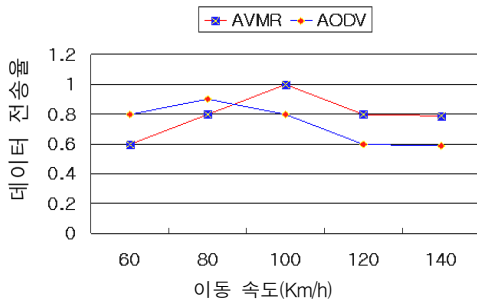
4. 모의실험 및 결과분석

본 논문에서 제안하는 액티브 네트워크 기반의 차량 간 멀티홉 라우팅 프로토콜의 구현은 NS2를 사용하였다. NS2는 버클리 대학에서 제작된 네트워크 시뮬레이션 툴로서 C++ 기반에 Tcl/tk로 개발되었다. NS2에는 상당히 발전된 단계의 멀티캐스트 MAODV 프로토콜이 이미 구현되어 있으므로 본 실험에서는 기존의 AODV 알고리즘과 본 논문에서 제안한 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜 알고리즘의 성능을 비교하는 시뮬레이션을 시행하고자 했다. AODV 프로토콜 실험과 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜을 실험하기 위해 시스템 OS는 레드햇 리눅스7 서버를 이용 하였으며, 시뮬레이션툴은 NS-ALLINONE2.31 버전으로 실험 환경을 구성하였다.



(그림 3) 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜 패킷 전달율

(그림 3)은 고속화도로에서 차량이 고속으로 이동하면서 근원지 노드에서 발신한 패킷을 전송하는 내용이다. X좌표는 전송시간을 50초로 정의하여 10초 간격으로 패킷을 발신하도록 정의하였다. Y좌표는 패킷 수신율로 0.2개의 간격으로 수신율을 표시하였다. (그림 3)에서 적색 그래프는 패킷이 성공한 수신율이며, 녹색의 그래프는 분실된 패킷 수신율이다. 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜은 실험결과 시간이 지연되어도 패킷 분실이 줄어들고 패킷수신율이 증가함을 알 수 있다.



(그림 4) 노드의 이동속도 변화에 따른 데이터 전송율

(그림 4)의 도표는 차량의 속도에 따른 데이터 전송율을 나타낸 것이다. 1을 기준으로 0에 가까

울수록 전송률이 좋지 않음을 나타내는데 AODV 프로토콜은 속도가 증가 할수록 전송률이 낮아져 신뢰도가 떨어짐을 보이고 있다. 반면에 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜은 AODV 프로토콜보다는 속도의 영향을 받지 않는 전송율을 보이고 있음을 알 수 있다.

본 논문에서는 단순히 패킷 라우팅 목적으로만 사용되었던 개념을 바꾸어 사용자가 자신의 사용 목적에 맞게 라우터의 기능을 변경할 수 있는 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜을 제안하였다. 제안된 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜 알고리즘 성능 평가를 위해 NS2를 사용하여 실험 환경 변수인 적용 영역, 노드 수, 차량 속도, 패킷 수 등을 이용하여 각 속도에서의 데이터 전송율과 클러스터 헤더 재 선출을 측정하기 위한 데이터 전송 지연시간을 비교, 분석하였다.

평균데이터 패킷 전달율은 트래픽 소스에서 발생한 데이터 패킷의 수와 최종 목적지에 도달한 패킷 수의 비율이며 이 비율은 라우팅 프로토콜의 정확한 데이터 패킷 전송능력을 정량화 할 수 있다. 평균 패킷전달 지연시간은 소스노드에서 데이터 패킷이 발생한 시점부터 경로상의 노드들을 경유하여 목적지 노드에 데이터 패킷이 도착한 시점까지의 평균 지연 시간을 지칭 한다.

데이터 전송률 측정에서는 전체적인 네트워크의 처리율은 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜이 AODV 프로토콜보다 약 2.2% 우수한 것으로 나타났으며, 전체 네트워크의 처리율은 큰 차이가 없었다. 그러나 180Km/h 미만의 속도에서는 액티브 네트워크기반의 차량 간 멀티홉 라우팅 방식이 약 6.2%의 우수한 성능을 보였다.

패킷전달 평균 지연 시간은 AODV 프로토콜의 경우 속도가 빨라지면 클러스터 헤더 이탈로 인한 클러스터 헤더 재 선출을 RERR 메시지를 발생시킨다. 그리고 발신지 노드에게 경로 탐색을 다시 하게 해야 하기 때문에 Active 패킷이 Active 노드에 알려 Data를 중단 시킨다. 이에 따라 Active 패

킷이 경로요청을 하여 빠른 재 선출이 이루어지는 액티브 네트워크기반의 차량 간 멀티홉 라우팅 프로토콜보다 성능이 저하됨을 볼 수 있었다.

5. 결 론

차량 간 통신은 고속으로 이동 중인 통신의 특성 때문에 일반적인 네트워크보다 훨씬 민감하다. 긴급 상황시 대처할 수 있는 방법에 치중해서 프로토콜을 연구한 결과 기존의 Ad-hoc 네트워크의 프로토콜인 AODV 프로토콜보다 네트워크의 성능을 보였다. 향후에는 데이터의 신뢰성 보장문제를 해결하기 위해 Active Network에서 많은 서비스를 제공할 수 있는 수동적인 네트워크보다는 새로운 기술이나 장치에 대해 유연하고 능동적인 Active Network의 연구가 진행 되어야한다.

참 고 문 헌

[1] 한국전산원, “웹 환경 구축 및 운영을 위한 보안 기술 연구”, NCA III-RER-97052, 1997.
 [2] 김신규, 한광택, “안전한 웹 응용프로그램 개발에 관한 연구”, 2004.
 [3] Nam-Deok Cho, Eun-ser Lee, Hyun Gun Park, “Security Intelligence : Web Contents Security System for Semantic Web”, KES, LNCS, ISBN 978-3-540-46537-9, Volume 4252/2006, pp. 819-828, 2006.
 [4] 한국전교육센터, “웹 어플리케이션 보안”, 2007.
 [5] Roger Fournier저 유해영 역, “웹 어플리케이션 개발 방법론”, 이한출판사, 2002.
 [6] Mike Shema McGraw-Hill Hacknotes, “Web Security Portable Reference”, Companies Inc, 2003
 [7] S. McClure, S. Shad, “Web Hacking : Attacks

and Defense”, Addison Wesley, 2003.
 [8] Saumil Shah, “One-way Web Hacking”, Addison Wesley, 2003.
 [9] 정보통신부 한국정보보호진흥원, 웹 어플리케이션 보안 템플릿, 2006.
 [10] 정보통신부 한국정보보호진흥원, 홈페이지 개발 보안 가이드, 2005.
 [11] 과학기술부, 새로운 방식의 공개열쇠 암호의 제작과 기존 방식의 공개열쇠 암호의 연구, 2002.



장혜숙

2000년 군산대학교 컴퓨터과학과 (석사)
 2008년 군산대학교 컴퓨터과학과 (박사)
 2000년 현재 군산대학교 컴퓨터과학과 시간강사

관심분야 : 무선네트워크, Ad-hoc, Active Network



이진관

1999년 군산대학교 컴퓨터과학과 (석사)
 2007년 군산대학교 컴퓨터과학과 (박사)
 2006년 현재 군산대학교 컴퓨터과학과(시간강사)

관심분야 : 센서네트워크, Ad-hoc, Active Network



정규철

1999년 군산대학교 컴퓨터과학과 (석사)
 2006년 군산대학교 컴퓨터과학과 (박사)
 1999년 현재 군산대학교 컴퓨터과학과 시간강사

관심분야 : 정보검색, 무선네트워크, Ad-hoc, Active Network



이 종 찬

1996년 숭실대학교 컴퓨터학과
(석사)

2000년 숭실대학교 컴퓨터학과
(박사)

2000년~2005년 한국전자통신 선
임연구원

2005년 현재 군산대학교 컴퓨터학과 교수
관심분야 : 모바일네트워크, 텔레매틱스



박 기 흥

1986년 숭실대학교 전자계산학과
(석사)

1995년 일본 토쿠시마대학
지능정보과학과(박사)

1987년 현재 군산대학교 컴퓨터
과학과 교수

2004년 현재 NURI사업 텔레매틱스 인력양성 사업단
(군산대) 단장

관심분야 : 자연어처리, 정보검색, 텔레매틱스