ns-2 기반 WAVE 멀티채널 시뮬레이션 프로그램 작성 및 검증

강우성* · 정진욱** · 진교홍***

Programming and Verification of WAVE Multi-Channel Simulation Program based on the ns-2

Woo-sung Kang* · Jin-uk Jung** · Kyo-hong Jin***

이 논문은 2010년도 창원대학교 연구비에 의해서 연구되었음

요 약

WAVE는 IEEE 1609.x 패밀리 표준들과 IEEE 802.11p 표준으로 구성된 차량 에드혹 네트워크의 대표적인 표준으로 차량 운행시 안전 서비스, 편의 서비스 등을 제공하기 위해 멀티채널 코디네이션과 채널 동기화와 같은 동작들을 포함하고 있다. 데이터 통신 및 네트워킹 기술들은 일반적으로 ns-2, OPNET, 그리고 OPNET++와 같은 시뮬레이터들을 사용하여 그 성능이 평가되지만, 현재까지 이러한 툴들은 WAVE 프로토콜의 동작들을 제공하고 있지 않다. 따라서 본 논문에서는 ns-2를 기반으로 하여 WAVE 멀티채널 시뮬레이션 프로그램을 작성하고 검증하였다.

ABSTRACT

A typical standard of vehicular ad hoc networks, WAVE which consists of IEEE 1609.x Family standards and IEEE 802.11p standard, includes the multi-channel coordination and channel synchronization function to provide safety services or public services during a car is driven. Generally, the performance of data communication and networking technologies is evaluated by using simulation tools, such as ns-2, OPNET, OPNET++, etc. However, these tools doesn't provide the operations of WAVE protocol. Therefore, in this paper, we implement and verify WAVE simulation program based on ns-2

키워드

차량 에드혹 네크워크, WAVE, 네트워크 시뮬레이터, 멀티채널 코디네이션

Key word

VANET, WAVE, ns-2 Multi-Channel Coordination

* 준회원 : 국립창원대학교 전자공학과

** 정회원 : 국립창원대학교 전자공학과

*** 정회원 : 국립창원대학교 전자공학과 (교신저자, khjin@changwon.ac.kr)

접수일자 : 2011. 02. 22

심사완료일자 : 2011. 03. 16

I. 서 론

차량 에드혹 네트워크(Vehicular Ad hoc Network)는 그림 1과 같이 컴퓨팅, 통신, 센싱 능력과 사용자 인터페이스(User Interface)를 갖추고 있는 차량과 기존의 인프라 네트워크와 연결되어 있는 노변 장치(Roadside Infrastructure)로 구성되며 차량 대 차량(V2V, Vehicle-to-Vehicle) 통신과 차량 대 인프라(V2I, Vehicle to Infrastructure) 통신 기술을 이용하여 운전자 및 승객에게 안전 서비스 및 다양한 멀티미디어 서비스를 제공할 수 있다[1].

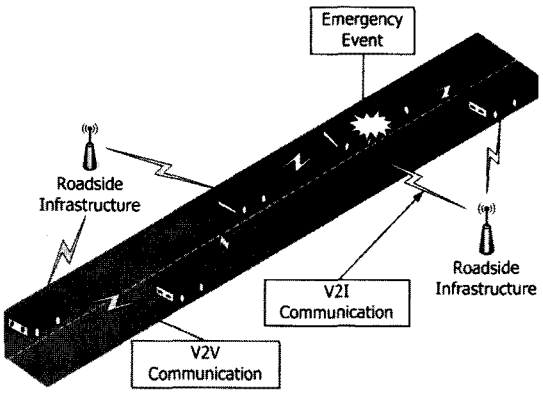

그림 1. 차량 에드혹 네트워크
Fig 1. Vehicular Ad hoc Networks

WAVE(Wireless Access in Vehicular Environment)는 대표적인 차량 에드혹 네트워크 통신 기술 표준으로 그림 2와 같이 IEEE 1609.1/2/3/4 표준과 IEEE 802.11p 표준으로 구성된다. IEEE 1609.1은 WAVE 어플리케이션에 대해서 기술하고 있으며 IEEE 1609.2는 시큐리티 서비스에 대해서 정의하고 있다. 또한 IEEE 1609.3은 네트워킹 서비스를 IEEE 1609.4는 멀티채널 코디네이션 동작을 각각 기술하고 있다. 한편 IEEE 802.11p는 고속으로 이동하는 차량에 적합한 물리 계층과 사용자 우선순위를 기반으로 하는 MAC 프로토콜을 정의하고 있다[2].

그 중에서 특히, IEEE 1609.4에 정의되어 있는 채널 코디네이션 알고리즘은 다수의 채널을 활용하는 WAVE 표준의 핵심 기술이다[3]. WAVE에서 제공되는 어플리케이션들은 지정된 채널로 메시지를 전송하게끔 설계되어지므로 관련 서비스를 제공받고자 하는 User는 자신이 원하는 서비스에 따라서 채널을 변경할 수 있어야 한다.

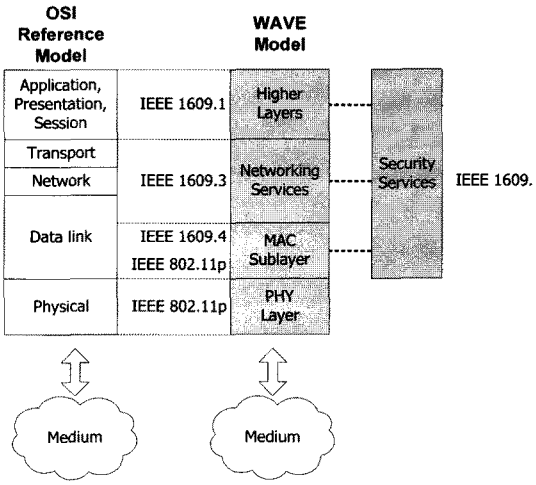

그림 2. WAVE 표준과 OSI 참조 모델의 비교
Fig 2. WAVE Standards and OSI Reference Model

한편 네트워크 구성요소들의 빠른 이동속도로 인한 토폴로지 변화가 극심한 차량 에드혹 네트워크의 경우에 테스트베드를 구현하여 성능을 평가하는 것은 현실적으로 매우 어렵다. 따라서 데이터 통신과 네트워크 분야에서 성능평가를 위해 일반적으로 사용되는 시뮬레이터를 사용해야 한다. 그러나 현재 활용되고 있는 시뮬레이터들은 차량 에드혹 네트워크 기술인 WAVE의 멀티채널 코디네이션 모듈을 포함하고 있지 않아 시뮬레이션이 불가능하다. 따라서 본 논문에서는 가장 많이 활용되는 오픈소스 기반의 시뮬레이터인 ns-2에 WAVE의 멀티채널 코디네이션 모듈을 구현하고 그 동작을 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 WAVE 기술에 대한 개요와 특징들을 간단히 기술하였다. 3장에서는 멀티채널, 채널 코디네이션 모듈의 구현 내용을 설명하고 4장에서는 동작을 검증한 결과를 설명한다. 마지막으로 5장에서는 결론 및 향후 과제를 제시하였다.

Ⅱ. WAVE 개요

그림 3과 같이 WAVE 디바이스들은 1999년에 DSRC(Dedicated Short Range Communication)용으로 5.8 ~ 5.9GHz 대역에 할당된 7개의 10MHz 채널들을 이용하여 통신하며, 이 채널들은 하나의 제어 채널(CCH: Control Channel)과 여섯 개의 서비스 채널(SCH: Service Channel)로 구성된다[4].

Channel Number	172	174	176	178	180	182	184
Channel Type	Service Channel	Service Channel	Service Channel	Control Channel	Service Channel	Service Channel	Service Channel
Frequency Range(GHz)	5.855 ~ 5.865	5.865 ~ 5.875	5.875 ~ 5.885	5.885 ~ 5.895	5.895 ~ 5.905	5.905 ~ 5.915	5.915 ~ 5.925

그림 3. WAVE에서 사용되는 채널들
Fig 3. Channels used in WAVE

WAVE 디바이스는 일반적으로 하나의 무선 송·수신기를 가지고 있기 때문에 제어 채널과 서비스 채널을 동시에 활용할 수 없다. 따라서 IEEE 1609.4는 채널 스위칭을 통해 WAVE 디바이스가 일정 시간동안 제어 채널과 서비스 채널을 번갈아 가면서 사용하는 멀티채널 코디네이션을 정의하고 있다. WAVE 디바이스들은 그림 4와 같이 제어 채널과 서비스 채널을 사용할 시점을 동기화한다.

시간은 Sync Interval(100msec)로 분할되며 이 Interval은 다시 CCH Interval(50msec)과 SCH Interval(50msec)로 나누어진다. 모든 WAVE 디바이스들은 CCH Interval에서는 제어 채널을 SCH Interval 동안에는 서비스 채널을 모니터링 해야만 한다[3].

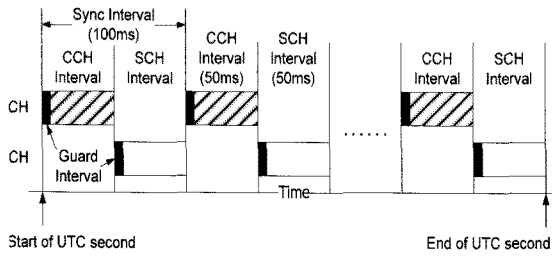

그림 4. WAVE 멀티채널 코디네이션
Fig 4. WAVE Multi-Channel Coordination

WAVE의 또 다른 특징은 MAC 프로토콜로 IEEE 802.11e의 EDCA(Enhanced Distributed Channel Access)를 사용한다는 것이다. EDCA는 CSMA/CA 기반의 MAC 프로토콜로 어플리케이션에 따라서 생성되는 패킷에 대해 다른 우선순위(Priority)를 부여할 수 있다[5]. 그림 5에서처럼 상위 계층으로부터 MAC 계층으로 전달된 패킷들은 자신의 우선순위와 대응되는 AC(Access Category)값을 가진 큐(Queue)에 저장된다. 큐의 헤드에 위치한 패킷들은 독립적으로 백오프(Backoff)를 수행한다. WAVE는 채널의 종류와 AC에 따라서 백오프에 사용되는 파라미터들의 값들을 다르게 설정함으로써 높은 우선순위를 가지는 패킷들이 먼저 전송될 수 있도록 하였다[3].

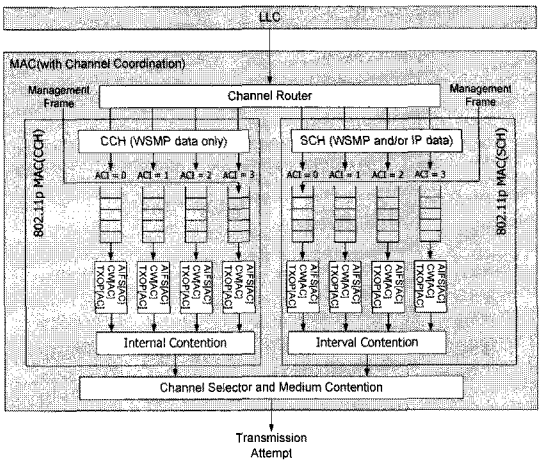

그림 5. 채널 코디네이션을 포함하는 WAVE MAC 아키텍처
Fig 5. WAVE MAC Architecture with Channel Coordination

Ⅲ. WAVE 멀티채널 시뮬레이션 프로그램 구현

3.1. WAVE 모바일노드 구조

ns-2에서는 무선 시뮬레이션을 지원하기 위해 기본적인 모바일노드 구조를 제공하고 있으며 Tcl로 작성되는 시뮬레이션 스크립트상에서 모바일노드의 구조를 직접 설정할 수 있다[6]. 그러나 기존 모바일노드의 구조를 사용하여 WAVE 멀티채널 코디네이션 시뮬레이션이 가

능하게 하려면 기존의 모듈들의 수정이 필요하고 멀티 채널 코디네이션 기능을 제공하는 새로운 모듈이 추가되어야 한다. 따라서 본 논문에서는 그림 6과 같이 현재 제공되고 있는 모바일노드 구조를 WAVE 모바일노드 구조로 변경하였다.

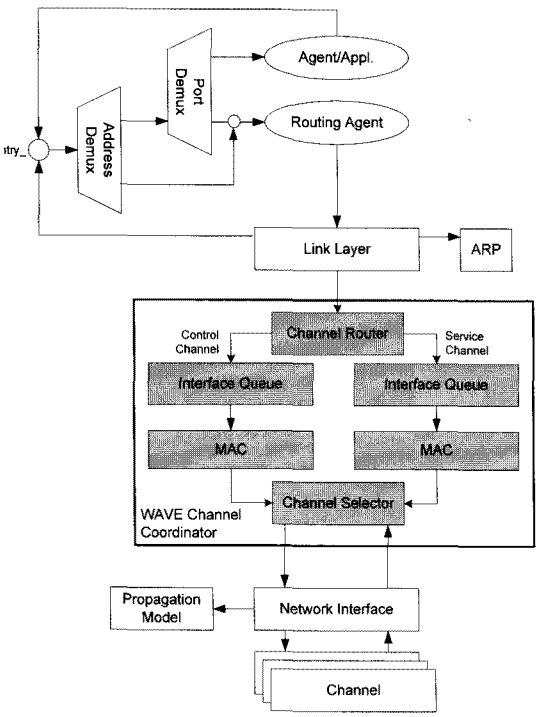

그림 6. 수정된 모바일노드 구조
Fig 6. The Structure of modified MobileNode

WAVE 모바일노드는 기존의 모바일노드 구조와는 다르게 WAVE Channel Coordinator를 가지고 있으며, 이는 Channel Router와 Channel Selector, 제어 채널로 전달되는 패킷을 처리하는 Interface Queue와 MAC 그리고 서비스 채널로 전달되는 패킷을 처리하는 Interface Queue와 MAC으로 구성된다.

3.2. WAVE Channel Coordinator 구현

시뮬레이션 프로그램에서 WAVE Channel Coordinator는 *WaveChannelCoordinator*라는 이름을 가지는 클래스이며 내부 구성요소들로부터 발생된 이벤트들을 처리하는 역할을 수행한다.

그림 7은 WAVE Channel Coordinator의 동작 원리를 나타낸 것이다. 본 논문에서는 Channel Router와 Channel Selector와 관련된 모듈들을 구현하고 추가하였으며 Interface Queue와 MAC은 TKN에서 개발한 EDCA 시뮬레이션 모델을 수정하여 사용하였다[7].

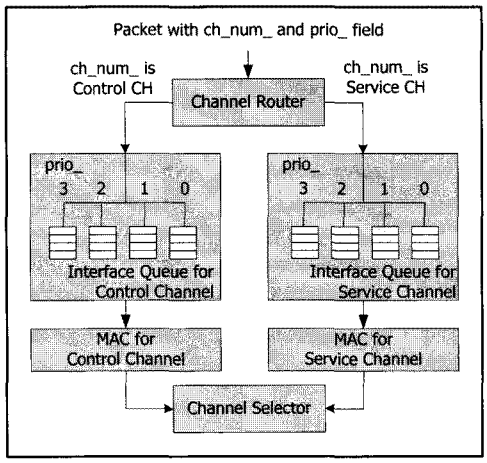

그림 7. WAVE Channel Coordinator의 구조
Fig 7. The Structure of WAVE Channel Coordinator

패킷이 전달되는 과정은 다음과 같다. 먼저 Channel Router는 LLC로부터 전달된 패킷의 헤더를 검사한 후 *ch_num_* 필드의 값에 따라서 제어 채널용 Interface Queue나 서비스 채널용 Interface Queue로 전달한다. 그 다음 Interface Queue는 전달된 패킷의 *prio_* 필드를 확인하여 4 개의 내부 큐들 중 하나에 저장하고 현재 인터벌(CCH Interval 또는 SCH Interval)에 따라서 사용 중인 채널의 상태를 확인하여 채널이 IDLE이면 큐의 헤드에 있는 패킷들은 백오프를 수행한다. 백오프 이후에 채널이 IDLE이면 패킷은 전송된다.

Channel Selector는 WAVE Channel Coordinator의 핵심으로 일정한 간격마다 채널을 변경한다. 이를 위해 그림 8과 같이 ns-2에 주기적으로 이벤트를 발생시키는 네 개의 타이머를 생성하였다. 타이머가 만료될 때 발생되는 이벤트는 *WaveChannelCoordinator*의 멤버 함수인 *wct_timer()*를 호출하며 *wct_timer()* 함수는 네 개의 타이머 중에서 어떤 타이머가 이벤트를 발생시킨 것인지 확인하고 채널을 변경한다.

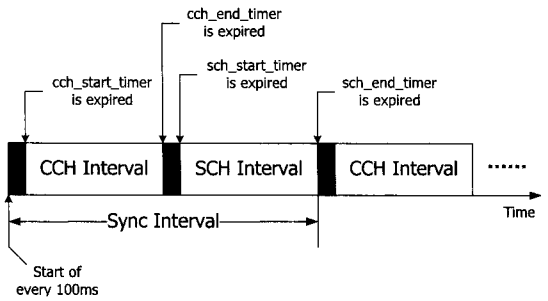

그림 8. 채널 인터벌과 이벤트 발생 시점
Fig 8. Channel Interval and an Event Time

표 1은 *wct_timer()*를 호출하는 타이머 및 이벤트의 종류와 수행하는 작업을 설명한 표이다. *cch_start_timer*와 *sch_start_timer*는 타이머가 만료하면 *wct_timer()*를 호출하여 각각 제어 채널(패킷의 *ch_num_* 필드 값에 해당하는 채널)과 서비스 채널로 채널을 변경한다. *cch_end_timer*와 *sch_end_timer*가 만료되면 채널을 7번으로 변경하도록 하였다. 7번 채널은 내부적으로 사용하는 채널이며 이 채널로는 통신이 이루어지지 않도록 처리하였다.

표 1. 네 가지 종류의 타이머
Table 1. Four Different Timers

타이머	이벤트	동작 결과
cch_start_timer	Expire	제어 채널로 채널 변경
cch_end_timer	Expire	제 3의 채널로 변경, 이 채널에서는 어떠한 통신도 수행되지 않음
sch_end_timer	Expire	
sch_start_timer	Expire	서비스 채널로 채널 변경

3.3. 다수의 채널 생성

앞서 말한 바와 같이 상위 계층 프로토콜(그림 6의 Agent)로부터 전달되는 패킷들이 모두 동일한 채널 번호를 가지도록 패킷의 헤더에 *ch_num_* 필드를 추가하였으며 Tcl 시뮬레이션 스크립트상에서 사용하고자 하는 채널을 지정할 수 있다. 즉, 특정 상위 계층 프로토콜로부터 WAVE Channel Coordinator로 전달되는 모든 패킷들은 동일한 채널이 지정된다.

ns-2는 무선 채널에 연결된 노드들을 리스트(List)로 관리하고 있다. 어떤 노드가 채널을 변경한다면 기존에 연결되어 있던 채널의 노드 리스트에서 노드를 제거하고, 새로 연결하려는 채널의 노드 리스트에 자신을 추가해야 한다. 그림 9와 같이 만약 MobileNode ②가 채널을 변경한다면, *WirelessChannel*의 멤버함수인 *removeNodeFromList()*를 호출하여 *prev_chan*로부터 자신을 제거하고 addNodeToList()를 호출하여 *next_chan*에 자신을 추가한다.

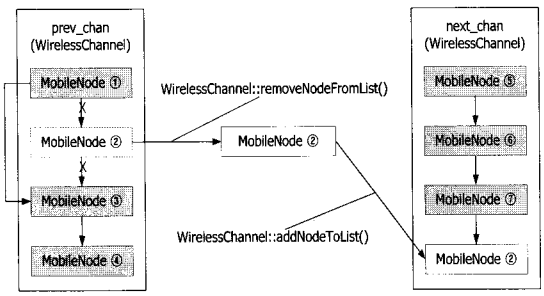

그림 9. 채널의 노드 리스트 관리
Fig 9. Management of Node List per Channel

Ⅳ. 시뮬레이션 프로그램 검증

개발된 WAVE 멀티채널 코디네이션 모듈의 유효성을 검증하기 위해서 두 개의 간단한 시뮬레이션들을 수행하였다.

[시뮬레이션] Channel Access Interval 확인

이 시뮬레이션의 목적은 개발된 WAVE 멀티채널 코디네이션 동작의 확인하는 것이다. 먼저, 그림 6과 동일한 구조를 가진 WAVE 모바일노드를 두 개 생성하고 다음과 같이 통신하도록 설정하였다. 먼저 한 노드(Node#0)는 무선 패킷을 송신하도록 설정하고 다른 한 노드(Node#1)는 무선 패킷을 수신하도록 설정한다. 그리고 Node#0는 CCH Interval과 SCH Interval에서 각각 채널 0번(*ch_num_*=0)과 채널 1번(*ch_num_*=1)으로 전송되는 패킷을 생성시키도록 한다.

시뮬레이션 수행 후에 생성된 트레이스(Trace) 파일로부터 표 2와 같은 결과를 추출할 수 있었다. 이 표는

패킷이 어느 채널에서 전송되며 생성되는 시간과 송신되는 시간을 나타내고 있다. 예를 들어, p2는 채널 번호 1(서비스 채널)으로 전송될 패킷으로 10.0300(CCH Interval)초에 생성된다. 그러나 p2는 서비스 채널 상에서 전송되어져야 하기 때문에 CCH Interval동안에는 전송되지 않고 SCH Interval이 시작되는 시점에 전송을 시도하게 된다. 따라서 10.0542초에 전송되는 것을 알 수 있다.

표 2. 각 패킷의 ch_num_ 및 생성/송신 시간
Table 2. ch_num_ and Creation/Transmission Time of Each Packet

패킷	채널 번호	생성(Agent)	송신(MAC)
p1	0 (CCH)	10.0200 (CCH Interval)	10.0201 (CCH Interval)
p2	1 (SCH)	10.0300 (CCH Interval)	10.0542 (SCH Interval)
p3	0 (CCH)	10.0700 (SCH Interval)	10.1042 (CCH Interval)
p4	1 (SCH)	10.0800 (SCH Interval)	10.0801 (SCH Interval)
p5	0 (CCH)	10.1200 (CCH Interval)	10.1201 (CCH Interval)
p6	1 (SCH)	10.1300 (CCH Interval)	10.1542 (SCH Interval)

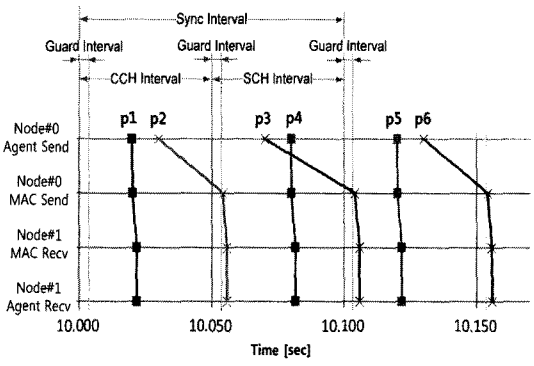

그림 10. Channel Access Interval 검증
Fig 10. Verification of Channel Access Interval

그림 10은 표 2의 내용을 그래프로 나타낸 것이다. p1, p4, p5의 경우에 전송될 채널의 번호와 패킷이 생성된 시점이 동일하여 현재 Interval에서 바로 전송되는 것을 알 수 있다. 반면에 p2, p3, p6은 현재 Interval에서 패킷을 전송할 수 없으므로 다음 Interval에서 전송되었다.

[시뮬레이션] 멀티채널 동작 검증

두 번째 시뮬레이션은 네트워크에서 동시에 여러 개의 채널을 사용할 수 있으며 서로 다른 채널에 연결된 노드들은 서로 간섭을 주지 않는다는 것을 확인하기 위해서 수행하였다.

시뮬레이션 시나리오는 다음과 같다. 먼저 WAVE 모바일노드를 4개 생성하고 각각 Node #1, #2, #3, #4이라고 명명하였다. Node #1, #2는 서비스 채널 1번을 사용하도록 설정하고 Node #3, #4는 서비스 채널 2번을 사용하도록 설정하였다. 임의의 시간에 Node #1과 Node #3이 패킷을 전송하고 어떤 노드가 패킷을 수신하는지 확인하였다.

그림 11은 시뮬레이션에서 각 노드의 패킷 송신 및 수신 이벤트를 발생 시간 순으로 나타낸 것이다. 예상한 바와 같이 0.06초에 Node #1이 전송한 패킷은 같은 서비스 채널 1번을 사용하는 Node #2가 수신하였음을 알 수 있다. 그리고 Node #3와 Node #4는 서비스 채널 2번을 사용하므로 Node #1이 보낸 패킷을 수신하지 못함을 확인할 수 있다. 반대로 0.064초에 Node #3이 전송한 패킷은 동일한 서비스채널 2번을 사용하는 Node #4만이 수신하였고, Node #1, #2는 서비스채널 1번을 사용하므로 수신하지 못하였음을 볼 수 있다.

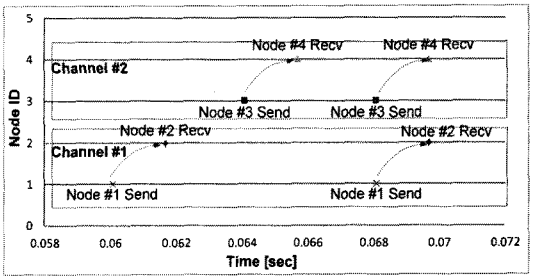

그림 11. 멀티채널 동작 검증
Fig 11. Verification of Multi-channel Operation

그리고 0.068초에는 Node #1과 Node #3이 동시에 패킷을 전송하였다. 이 패킷들은 서로 다른 채널에서 전송되므로, 패킷 충돌이 발생하지 않고 전송이 성공적으로 수행됨을 볼 수 있다. 이 결과를 통해서 각 채널이 독립적으로 동작하기 때문에 서로 다른 채널로 패킷을 동시에 전송할 수 있고 같은 채널에 연결된 노드 사이에서만 통신이 이루어짐을 확인할 수 있었다.

V. 결 론

본 논문에서는 기존의 시뮬레이션 툴에서는 제공되지 않는 WAVE 멀티채널 코디네이션 알고리즘을 ns-2에서 구현하였다. 구현된 프로그램을 두 가지 시뮬레이션을 통해 주기적으로 제어 채널과 서비스 채널간의 채널 변경이 정확하게 이루어지고 동시에 여러 개의 채널을 사용할 수 있음을 검증하였다.

본 논문에서 구현한 WAVE 멀티채널 코디네이션 시뮬레이션 프로그램은 WAVE와 관련된 연구뿐만 아니라 다른 무선 통신 기술의 멀티채널 시뮬레이션에도 많은 도움을 줄 것으로 예상된다. 향후 이 시뮬레이션 프로그램을 활용하여 WAVE MAC 프로토콜의 성능을 평가하고 분석할 것이다.

참고문헌

[1] 김태홍, "차량 통신 시스템: 기술, 응용 및 지능형 교통시스템의 전망," 한민족과학기술자네트워크, 분석자료, 2010년 1월, pp.090-096

[2] Morgan, Y, L., "Notes on DSRC & WAVE Standards Suite: Its Architecture, Design, and Characteristics," IEEE Communications Surveys & Tutorials, Issue 4, pp. 504-518, May, 2010

[3] IEEE 1609 WG, "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation," IEEE Std 1609.4, 2006

[4] Sebastian Grafling, Petri Mahonen and Janne Riihijarvi, "Performance Evaluation of IEEE 1609 WAVE and IEEE 802.11p for Vehicular Communication," Ubiquitous and Future Networks (ICUFN) 2010, pp. 344-348, June, 2010

[5] IEEE 802.11 WG, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements," IEEE Std 802.11e, November, 2005

[6] Network Simulator 2 Official Web Site, http://www.isi.edu/nsnam/ns

[7] TKN, An IEEE 802.11e EDCA and CFB Simulation Model for ns-2,", http://www.tkn.tu-berlin.de/research/802.11e_ns2

저자소개

강우성(Woo-sung Kang)

2007년 창원대학교 전자공학과 공학사
2011년 창원대학교 전자공학과 공학석사

※관심분야 : 컴퓨터 언어, 무선센서네트워크, VANET

정진욱(Jin-uk Jung)

2004년 동의대학교 멀티미디어 공학과 공학사
2006년 동의대학교 디지털미디어 공학과 공학석사

2008년 창원대학교 전자공학과 박사과정 수료

※관심분야 : 데이터 통신, 무선센서네트워크, VANET

진교홍(Kyo-hong Jin)

1991년 부산대학교 컴퓨터공학과 공학사
1993년 부산대학교 컴퓨터공학과 공학석사
1997년 부산대학교 컴퓨터공학과 공학박사
1997 ~ 2000년 국방과학연구소 선임연구원
2000 ~ 2004년 동의대학교 멀티미디어공학과 조교수
2004 ~ 현재 창원대학교 전자공학과 교수
※관심분야 : 데이터통신, 센서네트워크, 유비쿼터스 컴퓨팅, VANET