韓國컴퓨터情報學會 論文誌
第15卷 第2號, 2010. 2.

2010-15-2-4-4

MS 윈도우즈 운영체제 기반의 무선 네트워크 성능 분석 시뮬레이터의 설계 및 구현

최 관 덕*, 장 호**

Simulator for Performance Analysis of Wireless Network based on Microsoft Windows Operating Systems

Kwan-Deok Choi*, Ho Jang**

요 약

현실적으로 무선 네트워크의 정확한 성능 평가를 위해서는 다수의 노드들을 실제 네트워크 환경에 적용하여 노드들 간의 전송 데이터를 실시간으로 수집하여야 하는 어려움이 있다. 이러한 이유에서 공간 및 시간적인 제약사항을 극복함과 동시에 매체접근기법, 라우팅기법, 노드배치알고리즘 등과 같은 최적의 설계 방법을 찾기 위한 분석 도구가 필수적으로 요구된다. 이에 본 논문에서는 무선 네트워크 시스템을 효과적으로 시뮬레이션하기 위하여 멀티쓰레딩(multi-threading)기법을 응용한 시뮬레이션 도구의 구조와 설계방안을 제시하고 이를 구현하였다. 최종적으로 범용 네트워크 시뮬레이터인 NS2 무선 네트워크 모델에서 일반적으로 사용되는 무선 네트워크 환경 파라미터들을 구현된 시뮬레이터에 동일하게 적용하여 시뮬레이션을 수행하고 이를 비교함으로써 제안된 시뮬레이터의 유용성을 입증한다.

Abstract

To ensure accurate measurements of wireless network performance, it should be collected real-time data which are transmitted between a large number of nodes in the actual network environment. Therefore, it is necessary to develop simulation tool for finding optimal network system design method such as media access control, routing technique, ad-hoc algorithm of node deployment while overcoming spatial and temporal constraints. Our research attempts to provide an improved architecture and design method of simulation tool for wireless network is an application of multi-threading technique in these issues. We finally show that usability of the proposed simulator by comparing results derived from same test environment in the wireless LAN model of our simulator and widely used network simulation package, NS-2.

▸ Keyword : 무선 랜 시뮬레이션(Wireless LAN Simulation), IEEE 802.11 시뮬레이터(IEEE 802.11 Simulator), 멀티쓰레딩 처리기법(Multi-threading Technique)

• 제1저자 : 최관덕 교신저자 : 장호
• 투고일 : 2010. 01. 16, 심사일 : 2010. 01. 28, 게재확정일 : 2010. 02. 22.
* 대구과학대학 의무기록정보과 조교수 ** 영남대학교 컴퓨터공학과
※ 이 논문은 2009년도 대구과학대학 교육역량강화사업단의 지원을 받아 수행된 연구임.

I. 서 론

최근 무선 인터넷을 비롯한 무선 랜의 사용이 증가함에 따라서 무선 랜의 속도, 성능 및 QoS에 대한 보장성 등과 같은 서비스와 통신 성능에 대한 중요성도 함께 높아졌다. 결과적으로 유선 랜과 무선 랜이 공존하고 있는 현 시점에서 고려해야 할 과제 중 하나가 무선 랜 표준 안에 입각한 무선 랜 프로토콜(Wireless network protocol)에 대한 올바른 성능 분석이라고 할 수 있다. 이에 본 논문에서는 IEEE 802.11 MAC 프로토콜 표준안에 충실하도록 구현된 새로운 무선 랜 시뮬레이터를 구현하였다. 구현된 무선 랜 시뮬레이터는 이산사건 처리 기반 시뮬레이터(discrete event simulator)이며, 사용하기 쉽고 분석하고자 하는 프로토콜 계층의 조작과 응용이 쉬운 장점이 있다. 기본적으로 IEEE 802.11 표준안[1]에서 제시한 파라미터들을 준수하였으며, 다양한 802.11 계열의 확장 프로토콜들도 시뮬레이션을 수행할 수 있다. 또한 실제 무선 네트워크의 성능과 가장 유사한 결과를 얻기 위하여 프레임 전송 실패율, 랜덤 백오프 시간, 폴링 리스트 등과 같은 시뮬레이션 파라미터들을 채널의 상태와 단말의 수에 따라서 조정할 수 있도록 설계하여 실제 무선 네트워크의 현재 상태와 유사한 시뮬레이션이 가능하도록 설계하였다.

본 논문의 구성은 2장에서 관련 연구에 대하여 간단히 살펴보고, 3장에서는 제안하는 시뮬레이터의 구조를 체계적으로 설명하며, 4장에서 기존의 NS2를 이용한 시뮬레이션 결과와 제안하는 시뮬레이터의 결과를 비교하여 효용성을 입증한다. 마지막으로 결론과 향후 과제를 5장에서 서술한다.

II. 관련 연구

본 장에서는 대표적인 무선 네트워크 MAC 프로토콜인 IEEE 802.11의 메커니즘과 이산사건 시뮬레이션기법에 대하여 알아본다.

IEEE 802.11 표준 작업 그룹(standard working group)에 의하면 노드들의 매체에 대한 접근 제어 방법은 그림 1에서와 같이 MAC 부계층에 DCF(distributed coordination function) 방식과 PCF(point coordination function) 방식, 그리고 이들의 혼합 방식이 정의되어 있다. DCF 방식은 IEEE 802.11 MAC에서 사용되는 기본적인 접근 방법이며 이 방식은 경쟁기반의 서비스를 제공하며 접근 방법으로 랜덤 백오프 알고리즘(random backoff algorithm)을 사용한다. 반면, PCF 방식은 선택적으로 사용되는 방법으로서 AP(access point)와 같은 중재기에 의한 폴링(polling)기법을 이용하여 노드들의 채널 점유 방법을 관리한다[2].

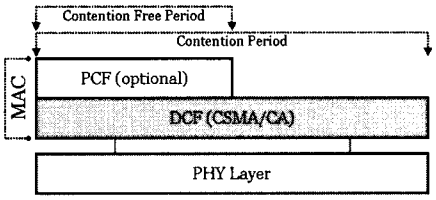

그림 1. IEEE 802.11의 MAC 구조
Fig. 1. Architecture of IEEE 802.11 MAC

DCF 방식에서의 백오프 알고리즘은 일정한 시간 간격(unit time)마다 전송을 위하여 선택한 슬롯의 번호를 감소시켜 그 값이 0이 될 때 전송을 시작하는 방법이다.

1. DCF에서의 랜덤 백오프 시간

경쟁 기반 서비스인 DCF 방식에서는 매체 최소 유휴(idle) 시간인 DIFS(distributed interframe space)를 이용하여 전송 매체의 가용 여부를 판단한다. 프레임 전송이 완료되고 DIFS가 경과된 후에는 경쟁 윈도우가 나타나게 되는데 이는 다수의 슬롯들로 나뉘며 슬롯 하나의 길이(SlotTime)는 매체 의존적이며 고속 물리 계층일수록 더 짧다. 매체 접근을 위하여 경쟁하는 노드들은 랜덤으로 슬롯 하나를 선택하여 일정 시간을 기다린 후 매체를 점유하게 되는데 이때 모든 슬롯의 선택 확률은 균등하다. 슬롯이 랜덤하게 선택됨으로써 백오프 타이머(backoff timer)는 $[random(i) \times SlotTime]$ 크기의 지연 시간을 발생시키며 이를 랜덤 백오프 시간이라고 한다.

이러한 백오프 알고리즘은 다중 노드 간의 매체 점유를 위한 경쟁 상황에서 상호간 통신을 주고 받을 수 없는 노드들 간의 충돌을 미연에 회피하여 이를 최소화하기 위한 방법으로 사용된다[2].

2. DCF를 이용한 경쟁 기반 접근 방법

일반적으로 그림 2에서 설명하는 바와 같이 DCF 접근 환경 하에서는 DIFS 기간 이상으로 매체의 유휴상태가 지속되면 전송이 임박한 노드의 PLCP (physical layer convergence procedure) 헤더를 포함하지 않은 프레임, 즉 MPDU(MAC protocol data unit)부터 즉시 전송할 수 있다.

만약 이러한 조건에서 반송파 감지 메커니즘에 의하여 매체가 사용 중이라고 감지되었을 경우에는 랜덤 백오프 알고리즘에 의해서 경쟁 윈도우의 크기가 바뀌게 되며(보통의 경우

에는 이전 단계의 2배가 된다), MPDU를 전송하거나 혹은 재전송하게 되더라도 MPDU와 ACK 신호의 전송을 허락하기에 시간이 충분하지 않다면 노드는 선택된 랜덤 백오프 시간에 의하여 전송을 연기하게 된다[2][3].

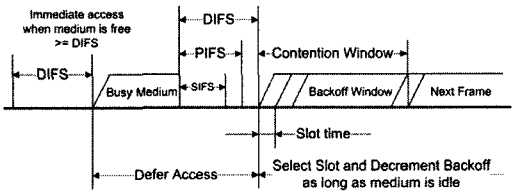

그림 2. IEEE 802.11에서 DCF를 이용한 접근 방식
Fig. 2. Media Access Control Technique of IEEE 802.11 DCF

3. DCF에서의 백오프 절차

DCF 방식에서 이루어지는 백오프 절차는 전송 노드가 전송에 실패하였다고 추측될 때 실시된다. 모든 백오프 절차는 매체가 DIFS 기간 동안에 대하여 이상적(유휴상태)이라고 판단될 때 수행된다. 만약 매체가 유휴상태에 있음이 감지된다면 백오프 동작은 슬롯타임에 의해 백오프 시간을 줄이는 절차를 수행한다. 반면에 매체가 백오프 과정 동안 사용 중이라고 판단되면 백오프 시간은 줄어들지 않고 멈추어 있다. 최종적으로 데이터 전송은 백오프 타이머가 0이 될 때마다 시작된다[2]. 그림 3은 DCF 백오프 절차에 의해 5개의 노드들이 경쟁하는 상황을 보여주고 있다.

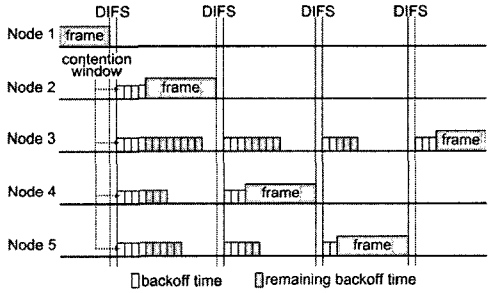

그림 3. DCF에서의 백오프 절차
Fig. 3. DCF Backoff Procedure

이러한 백오프 과정을 통하여 다중 노드들이 전송을 연기하거나 랜덤 백오프 동작으로 진입할 때 마다 랜덤 함수에 의해 결정된 백오프 시간을 선택하게 된다. 이전의 사실로 부터 노드들이 매체 접근 경쟁에서 이겨 데이터 전송에 성공하기 위한 기본 조건은 최소의 백오프 시간을 획득하는 것임을 알 수 있다. 결과적으로 경쟁에 참여한 다수의 노드들 중에서 충돌을 최대한 피할 수 있는 유일한 방법은 중복되지 않은 백오프 시간을 할당 받는 것이라고 볼 수 있다.

4. PCF를 이용한 전송과정

PCF 구간은 IEEE 802.11 표준의 옵션 부분으로 경쟁기반 전송방식과 달리 비경쟁 프레임을 전송한다. 인프라스트럭처(infrastructure) 네트워크에 포인트 조정자(point coordinator) 역할을 할 수 있는 AP가 존재한다면 이 포인트 조정자의 통제 하에서 폴링방식으로 전송권을 얻게 된다. 따라서 PCF에서는 RTS, CTS를 사용하지 않는다. 포인트 조정자의 폴(poll)은 CF-Pollable 스테이션만이 전송받을 수 있는데, 이 스테이션들은 인증(authentication)과 결합(Association) 과정을 거쳐 폴링리스트에 등록이 된 스테이션들에게만 전송권이 주어진다. PCF 전송과정은 다음과 같다[1].

포인트 조정자는 DIFS보다 더 적은 시간 안(PIFS)에 매체를 점유하여 전송권을 얻는다. 기본 서비스 셋(BSS) 내에 있는 스테이션들은 비컨이 전송되도록 스케줄링된 시간인 TBTT(target beacon transmission time)에서 NAV(network allocation vector)를 CFMaxDuration으로 설정하여 폴이 되지 않은 여러 다른 스테이션들(hidden station)의 전송을 막는다. 데이터 전송에 대한 ACK는 피기백(piggyback)되며, 만약 non-CF-Pollable 스테이션이나 폴링 대상에서 제외된 CF-Pollable 스테이션의 ACK 프레임은 무경쟁구간(CFP) 동안에 DCF ACK 과정을 이용하여 ACK 프레임을 전송한다. 포인트 조정자는 무경쟁구간의 시작시점에서 일단 매체의 상태를 검토한다. 매체가 PIFS 구간동안 유휴상태이면 전송을 하게 되는데 만약 매체가 점유(busy)상태인 경우에는 매체가 유휴상태가 될 때까지 기다린 후 비컨 프레임을 전송한다[2].

5. 이산사건 시뮬레이션기법

제안하는 무선 랜 시뮬레이터는 이산사건 처리기법을 바탕으로 구현되었다. 순차적으로 진행되는 시뮬레이션이 객체중심의 시스템 내에서의 물리적인 진행에 초점이 맞춰진 반면, 이산사건 시뮬레이션은 논리적인 시스템의 상태와 사건의 발생으로 인한 상태의 변화에 따라서 논리적으로 진행된다는 것에 큰 차이가 있다. 다시 말하면, 그림 4에 나타난 바와 같이 이산사건 시뮬레이터에서의 주체는 사건이 된다. 사건은 상태의 주체가 되는 객체에게 상태적인 변화를 일으키도록 하는 주요인이다. 상태는 입력 사건에 의하여 변화된 출력 사건을 내보내며, 이러한 사건들이 사건 목록을 구성하게 된다. 또한 상태는 임의의 사건에 의하여 특정한 이산 시점에서 변화를 일으키며, 이로 인하여 상태의 변화가 발생하면 이에 따른 시간의 경과가 타임스탬프에 표시된다. 즉, 전체의 시뮬레이션은

이산형의 시점들로 이루어진 시간 전개에 따라서 진행된다[4].

일반적으로 이산사건 기반의 시뮬레이터는 철저한 현상 규명에 집착하는 일반적인 시뮬레이터에서 발생하기 쉬운 처리시간에 대한 오차를 줄일 수 있다는 장점이 있다[5]. 실제로 객체중심으로 시뮬레이션을 진행하는 시뮬레이터에서는 객체 상태에 변화가 발생하여 데이터를 처리하는 시간뿐만 아니라 실제로는 동작하지 않으나 부수적으로 경과된 시간까지도 포함되어 있다. 이에 비하여 이산사건 시뮬레이터는 사건 발생에 따른 시간의 경과를 표시하므로 논리적인 처리시간에 따른 통계적 데이터를 정확하게 얻을 수 있으며, 논리적인 시뮬레이션의 전개로 인하여 시뮬레이션 결과에 대한 사전 유추 작업이 가능한 것이 특징이다.

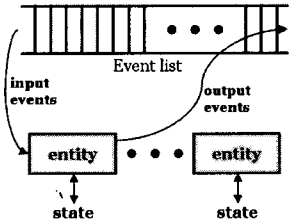

그림 4. 이산사건 시뮬레이션의 개념도
Fig. 4. Concepts of Discrete Event Simulation

III. 무선 랜 시뮬레이터의 설계

1. 구현된 시뮬레이터의 구조

이산사건 기반의 구현된 시뮬레이터는 보편적으로 사용되는 컴퓨터 언어인 C와 C++로 구성되어 있으므로, 이해가 쉽고 조작과 변형이 용이하며 다른 상위 또는 하위계층의 프로토콜 시뮬레이터와 연동이 가능하도록 설계되어 있다.

그림 5에서와 같이 구현된 시뮬레이터의 동작 구조는 크게 두 부분으로 나눌 수 있는데, 시뮬레이션에 사용될 각종 파라미터들을 이용하여 전체 시뮬레이션과정을 수행하는 시뮬레이션엔진 모듈과 사용자로부터 시뮬레이션에 필요한 파라미터를 입력받기 위한 사용자인터페이스 모듈이다.

구현된 시뮬레이터의 핵심부인 시뮬레이션엔진 모듈은 다음과 같은 내부 클래스들로 이루어져 있으며, 이들 클래스는 최종적으로 시뮬레이터 클래스와 유기적으로 결합되도록 마이크로소프트 비주얼 C++를 이용하여 ActiveX COM DLL 실행코드로 제작되었다. 우선적으로 시뮬레이션 수행에 앞서 노드 클래스모듈은 경쟁윈도우의 크기 및 백오프 카운터 값을 설정하며 백오프 동작을 초기화한다. 다음으로 소스 클래스 모듈에서 패킷의 크기와 발생 시간 간격을 조정하고, 시뮬레이션에 사용되는 전체 패킷의 구조를 패킷 클래스 모듈에서 정의한다. 또한, 패킷 발생 분포 특성과 전체 노드의 적용 범위와 같이 시뮬레이션의 전체 과정에 영향을 미치는 특성 값들은 파라미터 클래스 모듈에 기술되어 있다. 최종적으로 이러한 환경 변수들을 기초로 무선네트워크 물리계층의 종류, 노드 발생률, 경쟁 노드들의 전송 큐 관리, 노드 경쟁 메커니즘, 전송 노드의 처리와 경쟁에 실패한 노드의 재진입 방법 등을 주 실행부인 시뮬레이터 클래스 모듈에서 결정한다. 결과적으로 전체 시뮬레이션 과정은 사용자가 인터페이스 모듈에서 입력한 시뮬레이션 파라미터를 시뮬레이션엔진 모듈에서 수합하고 이를 기반으로 모델링된 무선 네트워크 환경을 정해진 시간동안 멀티쓰레딩 기법으로 시뮬레이션을 수행하도록 구현하였다. 부가적으로 시뮬레이션이 종료된 후에는 저장 기능 모듈에 의하여 그 결과가 텍스트 파일로 출력되어 저장된다.

이와 함께, 마이크로소프트 비주얼 베이직으로 작성된 인터페이스 모듈은 사용할 파라미터들을 사용자로부터 입력받기 위한 사용자인터페이스와 입력받은 파라미터들을 시뮬레이션엔진의 메인 클래스로 전달하여 전체 시뮬레이션 과정을 수행하는 제어부로 이루어져 있다.

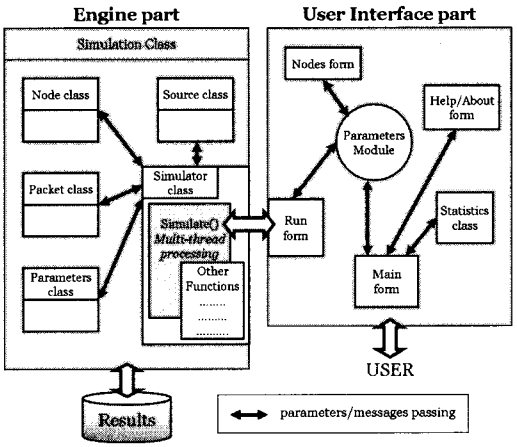

그림 5. 구현된 무선 랜 시뮬레이터의 동작 구조
Fig. 5. System Architecture of Proposed Wireless LAN Simulator

시뮬레이터를 구현하는 과정에서 무엇보다도 고려해야 할 사항은 전체 시뮬레이션과정에서 동시다발적으로 진행되는 병행 프로시저들 간에 발생하는 간섭현상을 최대한 억제하는 것이다. 이러한 간섭현상은 특정 프로시저가 의도하지 않게 진행되지 않아 전체 시뮬레이션과정이 원활하지 않게 되는 결과를 초래할 수 있다[6]. 이에 본 논문에서는 각 프로시저에

독립적인 쓰레드를 할당하여 전체 시뮬레이션과정이 진행되는 동안에 프로시저들의 일시적인 멈춤 현상을 최대한 제거하였다. 이는 마이크로소프트 운영체제의 멀티쓰레딩 기능을 최적화시킨 쓰레드제어모듈을 마이크로소프트 비주얼 베이직으로 작성하여 시뮬레이터엔진과 함께 탑재시킴으로써 극복하였다.

2. 구현된 시뮬레이터의 특징

앞서 살펴본 멀티쓰레딩을 지원하는 이산사건 처리기반의 시뮬레이터는 IEEE 802.11 프로토콜을 비롯하여 확장된 프로토콜들(IEEE 802.11a, IEEE 802.11b, IEEE 802.11g) 대부분을 시뮬레이션을 수행할 수 있는 진보된 설계방식의 무선 랜 시뮬레이터다. 이는 기본적으로 IEEE 802.11 표준안을 기초하여 매체접근제어(MAC) 프로토콜에 대한 동작특성 시뮬레이션을 세밀하게 수행할 수 있으며, 다음과 같은 다양한 특성을 가지고 있다.

각 노드의 거리에 따라 변화하는 데이터 전송률에 빠르게 반응한다.

은닉 터미널(hidden terminal) 문제의 분석 기능을 제공한다.

모든 노드가 IEEE 802.11g 프로토콜을 사용하는 경우뿐만 아니라 IEEE 802.11b 프로토콜이나 이전버전의 프로토콜을 사용하는 일부 노드에 대한 시뮬레이션 파라미터 설정이 가능하다.

RTS/CTS외에도 네트워크 성능 향상을 시키기 위해 IEEE 802.11g에서 새롭게 정의한 CTS-to-Self 가상 반송파 감지 및 보호 메커니즘(virtual carrier sense and protection mechanism)을 지원한다.

MAC 계층의 성능과 관련된 다양한 시뮬레이션 결과(패킷 전송률, 네트워크 이용률, 매체접근 지연시간, 큐 지연시간, 패킷전송 전체 지연시간, 지터, 평균 재전송율, 평균 패킷 큐 길이 등)에 대한 통계치를 제공한다.

이러한 특성들 외에도 운영체제 플랫폼으로 보급률이 높은 마이크로소프트 윈도우즈(MS-Windows 95/NT 계열 및 XP 계열)를 채택함으로써, 네트워크 성능 측정을 위한 추가적인 시스템 구축비용이 절약되고, 무선 랜 메커니즘 및 성능 분석에 필요한 파라미터 설정 방법이 용이하며, 추가적으로 프로토콜 메커니즘을 수정할 수 있는 다양한 기능들을 제공하기 때문에 교육용 시뮬레이터로 사용될 수 있는 장점을 갖추고 있다.

IV. 성능 평가

본 장에서는 동일한 시뮬레이션 파라미터를 이용하여 구현된 무선 랜 시뮬레이터와 NS2 버전 2.31[7]의 무선 랜 모델에서 각각 시뮬레이션을 수행한 결과를 비교 분석함으로써 본 논문에서 제안한 시뮬레이터의 효용성을 확인하였다.

실험은 표 1에 정리된 파라미터 값들을 바탕으로 펜티엄 4 3.0 GHz CPU를 이용하여 수행하였다.

표 1. 시뮬레이션 파라미터
Table 1. Simulation parameters

파라미터	설정 값
노드 수	20개 (5개의 802.11b 노드 포함)
데이터전송률	54 Mb/s
평균패킷크기	8 Kb (일양분포)
패킷생성빈도	8 packets/s (푸아송분포)
MAC 접근모드	RTS/CTS
물리계층	IEEE 802.11g/b 혼합
적용범위(x, y)	100m × 200m
시뮬레이션시간	600초

전체 실험에서는 공통적으로 RTS/CTS를 활성화시키고, 데이터 패킷을 8000 바이트, 랜덤 시드 값을 100으로 설정한 동일한 조건에서 비교대상인 두 종류의 시뮬레이터를 이용하여 600초 동안 시뮬레이션한 후 추출된 100개의 결과 값을 서로 비교하였다. 그 결과, 전체 실험을 통하여 본 논문에서 구현한 시뮬레이터에서 측정된 성능 결과들은 NS2에서 나타난 그것들과 비교하여 큰 차이가 없음을 확인하였다.

시뮬레이션을 수행하기에 앞서 구현된 시뮬레이터를 이용할 경우에는 시뮬레이션에 필요한 기본적인 설정 값들을 손쉽게 수정할 수 있으며, 동시에 시뮬레이션 결과 값들을 편리하게 확인할 수 있다. 이는 비교대상인 NS2 시뮬레이터는 텍스트기반의 고전적인 사용자 인터페이스를 채택한 반면에, 본 논문에서 설계한 무선 랜 시뮬레이터에서는 그림 6과 같이 메인 실행화면을 비롯한 대부분의 인터페이스 모듈을 그래픽기반의 사용자 인터페이스로 구현하였기 때문이다.

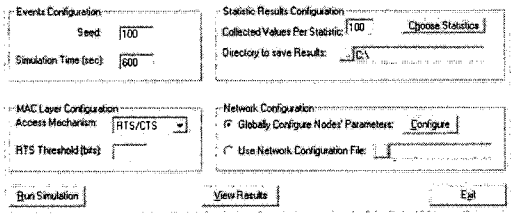

그림 6. 시뮬레이션 사용자 인터페이스 화면
Fig. 6. User Interface Screen of Simulation

그림 7은 시뮬레이션에 적용될 노드들의 수와 물리계층의 종류, 패킷의 크기 및 발생 빈도, 그리고 데이터 전송률 등과 같은 다양한 파라미터들을 사용자가 직접 설정할 수 있는 보조 인터페이스화면이다.

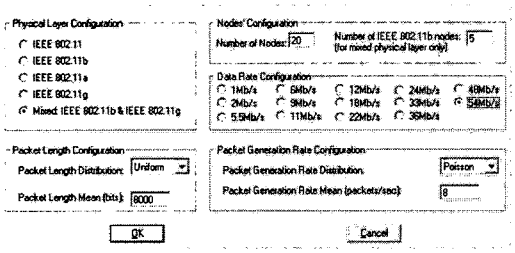

그림 7. 노드 파라미터 설정 화면
Fig. 7. Configuration of Node Parameters

본 논문에서 수행한 비교실험에서는 먼저 네트워크 부하를 가변시켜 가면서 무선 매체의 가용 한도가 포화 상태에 이르게 한 다음, 구현된 시뮬레이터에서의 전체 패킷 처리량과 NS2 시뮬레이터에서의 전체 패킷 처리량을 비교하였다. 또한 실제의 이벤트 기반 네트워크 환경에서 의 노드들의 동작 상황을 확인하기 위하여 실험에서는 정상상태(steady state)의 무선 랜 환경에서 동시에 발생되는 패킷들의 발생 빈도를 푸아송분포로 모델링한 뒤 이때의 패킷 처리량을 각각 측정하였다.

그림 8에서 보는 바와 같이 구현된 시뮬레이터에서 측정된 전체 패킷 처리량이 어느 특정 노드에 집중되지 않고 고른 분포를 보이면서 기존의 NS2에서 추출된 결과와 비교하여 전반적으로 비슷한 것을 알 수 있다.

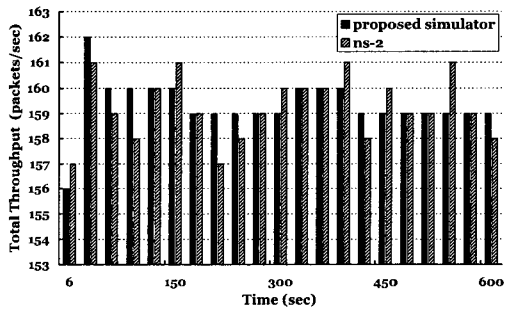

그림 8. 전체 패킷 전송률 비교
Fig. 8. Comparison of Total Packet Throughput

그림 9의 결과는 각 노드들이 매체접근을 시도하는 경우에 발생하는 지연시간의 평균값을 나타내고 있다. 결과에서는 본 논문에서 제안된 구조의 시뮬레이터로부터 측정된 평균 지연시간이 NS2 시뮬레이터와 비교하였을 때 미세하지만 좀 더 작은 값으로 측정되었다. 이는 시뮬레이터 엔진을 구현하는데 사용한 프로그래밍 언어의 연산규칙 및 데이터표현의 방식차이에서 기인한 것으로 분석된다.

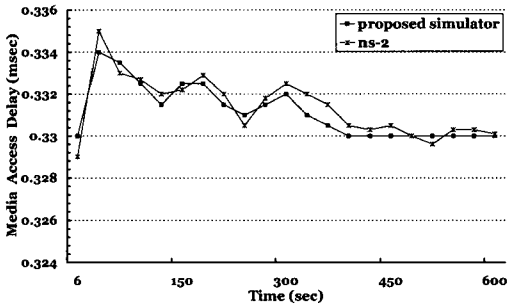

그림 9. 매체접근의 평균 지연시간
Fig. 9. Average of Media Access Delay

마지막 실험에서는 푸아송분포에 기반을 두어 발생되는 이벤트 부하에 대하여 무선 랜의 평균 지연시간에 대한 변화량(지터)을 측정하여 보았다. 실험에서 각 시뮬레이터로부터 측정된 전체 노드들의 전송 지연시간의 평균 변화량은 그림 10에 나타나 있다.

이때 지연 시간의 증감에 대한 변화량을 보다 정확하게 측정하기 위해 각 노드에 탑재되는 소프트웨어 시스템에 의한 지연이나 하드웨어의 전자 특성에 의한 편차 등을 일정량씩 고려하였다. 실험에서 이러한 특성을 반영하기 위하여 각 노드의 패킷 전송시간을 0ms에서 1ms 까지 범위의 랜덤 값을 더하여 확장하였다[8].

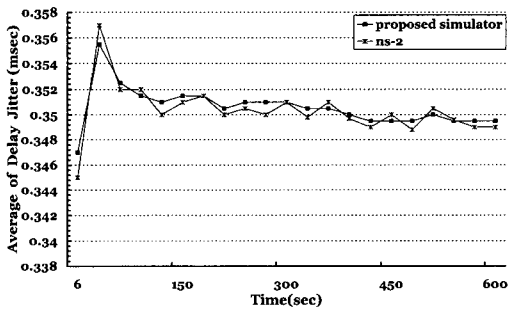

그림 10. 전송 지연시간의 평균 변화량
Fig. 10. Average of Delay Jitter

일반적으로 무선 랜의 동작 초기에는 지연시간에 대한 지터값의 변동 폭이 상대적으로 크게 나타나다가 시간이 점점 흐를수록 작아지는데, 시뮬레이션 결과에서도 동작시간이 흐를수록 무선 랜의 안정성이 전체적으로 높아짐을 보여 주고 있다[9].

V. 결론 및 향후 과제

본 논문에서는 무선 랜을 구성하는 노드들의 정보 전송 메커니즘을 효율적으로 분석하기 위한 시뮬레이션기법을 설계하고 구현하였다. 구현된 시뮬레이터 엔진의 장점은 운영체제에서 제공하는 멀티쓰레딩기법을 응용하여 각 노드 간의 상호 동작에서 발생될 수 있는 측정 오차를 최대한 줄이고, 전체 시뮬레이션 과정을 이산사건 방식으로 모델링하여 병행 처리함으로써 현재 진행되는 시뮬레이션 태스크가 수반되는 다른 태스크에 의해 예기치 않게 보류(suspend)되는 현상을 가급적 억제시키는 것이다[10][11]. 이러한 효용성을 입증하기 위해 본 논문에서는 동일한 조건의 무선 랜 모델을 제안한 시뮬레이터와 신뢰성을 공인받고 있는 네트워크 시뮬레이터인 NS2에 함께 적용하여 도출된 시뮬레이션 결과를 비교분석하였다. 구현된 무선 네트워크 시뮬레이터에서 측정된 성능요소들은 NS2에 비하여 약간의 오차가 발생했지만 이는 거의 무시되는 정도라고 볼 수 있다.

최종적으로 본 논문에서 제안한 시뮬레이터는 고가의 무선 네트워크 시뮬레이션 도구를 대신하여 교육현장에서 충분히 활용될 수 있을 것으로 전망된다.

앞으로 구현된 시뮬레이터의 성능 개선을 위해서 시뮬레이션 파라미터의 정밀도를 높임과 동시에 이벤트 기반의 현실적인 무선 랜 환경을 원활하게 시뮬레이션하기 위한 지원도구 모듈과 측정된 성능 평가 결과를 평가 더 쉽게 비교분석하기 위한 그래프 출력 모듈을 포함한 추가적인 라이브러리 구축 작업이 수행되어야 할 연구과제다.

참고문헌

[1] IEEE Std. 802.11-1999, Part 11, "*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*," Reference number ISO/IEC 8802-11:1999(E), IEEE Std. 802.11, 1999 edition, 1999.

[2] M. Gast, "*802.11 Wireless Networks: The Definitive Guide*," 2nd ED., O'Reilly, Apr. 2005.

[3] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE Journal on Selected Areas in Communication, Vol. 18, No. 3, pp. 535-547, Mar. 2000.

[4] 성영락, 김탁곤, 박규호, "병렬분산 이산사건 시뮬레이션," 한국정보과학회지, 제13권, 제4호, 20-32쪽, 1995년 4월.

[5] 윤종원, 정진우, "멀티미디어 서비스를 위한 IEEE 802.11 WLANs 기반의 무선 네트워크 동기화 알고리즘," 한국컴퓨터정보학회논문지, 제 13권 제 6호, 225-232쪽, 2008년 11월.

[6] D. Lee, P. Crowley, J. Baer, T. Anderson, and B. Bershad, "Execution Characteristics of Desktop Applications on Windows NT," isca, 25th Annual International Symposium on Computer Architecture (ISCA'98), pp. 27-38, 1998.

[7] USC ISI, "*ns-2 Notes and Documentation*," 2007.

[8] 장호, 정원석, 이기동, "실시간 무선 센서 네트워크에서 전송 지연 감소를 위한 MAC 개선 방안," 한국통신학회논문지 제34권 제8호(무선통신), 600-609쪽, 2009년 8월.

[9] 장호, "*분산 실시간 무선 센서 네트워크의 전송지연 감소를 위한 MAC의 개선 방안*," 영남대학교대학원 박사학위논문, 2008년 8월.

[10] 이승우, 구본철, 민홍, 이상호, 김용태, 허준영, 조유근, 홍지만, "센서 네트워크 시뮬레이터의 관련 연구 조사," 제10회 차세대 통신 소프트웨어 학술대회 (NCS'2006), 460-464쪽, 2006년.

[11] 김영곤, 김성진, 박찬익, "리눅스의 실시간 성능 향상을 위한 적응적 커널 쓰레드 제어 기법," 한국정보과학회 2009 한국컴퓨터종합학술대회 논문집 제36권 제1호(A), 302-307쪽, 2009년 6월.

저 자 소 개

최 관 덕

1989: 영남대학교 전산공학과 공학사
1991: 영남대학교 전산공학과 공학석사
2008: 영남대학교 전산공학과 공학박사
1996 - 2008: 대구과학대학 컴퓨터계열 조교수
2008 - 현재: 대구과학대학 의무기록정보과 조교수
관심분야: 컴퓨터시스템, 영상처리, 생물정보학 등

장 호

1994: 영남대학교 전산공학과 공학사
1996: 영남대학교 전산공학과 공학석사
2008: 영남대학교 컴퓨터공학과 공학박사
2001 - 2005: 구미1대학 컴퓨터전공 전임강사
2005 - 현재: 영남대학교 컴퓨터공학과 강사
관심분야: 실시간운영체제, 센서네트워크