

# 멀티홉 무선 센서 네트워크에서 버스트 데이타의 효율적인 전송을 위한 프로토콜에 관한 연구

(A MAC Protocol for Efficient Burst Data Transmission in  
Multihop Wireless Sensor Networks)

노 태 호 † 정 광 수 ††

(Taeho Roh) (Kwangsue Chung)

**요 약** 작은 센서 노드로 구성되는 무선 센서 네트워크는 멀티홉으로 무선 통신을 하는 주요한 특징을 가지고 있다. 지금까지의 응용은 주기적인 형태의 비교적 단순한 센싱 데이타를 취급하는 것이 대부분이었다. 하지만 최근 새로운 형태의 일시적이고 연속적인 버스트 데이타를 멀티홉으로 전송하는 좀 더 복잡한 응용들이 대두되고 있다. 따라서 이러한 응용을 효율적으로 지원하기 위한 전송 프로토콜에 관한 연구가 필요하다.

본 논문에서는 멀티홉 환경에서 버스트 데이타의 효율적인 전송을 위한 PIGAB(Packet Interval Gap based on Adaptive Backoff) 프로토콜을 제안하였다. 경쟁 기반 프로토콜인 PIGAB은 근원지 노드에서 동작하는 PIG(Packet Interval Gap) 제어 알고리즘과 릴레이 노드에서 동작하는 MF(MAC-level Forwarding) 알고리즘으로 구성되며, 새롭게 제안된 AB(Adaptive Backoff), CAB(Collision Avoidance Backoff), 그리고 UB(Uniform Backoff)를 기반으로 동작한다. 제안된 PIGAB 프로토콜은 이러한 알고리즘과 기법을 통해 감춰진 노드의 전송 시기를 인지하여 패킷마다의 전송 시기를 조절함으로써 멀티홉 환경에서 겪는 기본적인 문제를 해결할 수 있다. 시뮬레이션과 실제 실험을 통해 PIGAB 프로토콜이 기존 방식에 비해 멀티홉 환경에서 버스트 데이타를 안정적이고 신속하게 전송하는 것을 확인할 수 있었다.

**키워드** : 멀티홉 무선 센서 네트워크, 버스트 데이타, 백오프

**Abstract** Multihop is the main communication style for wireless sensor networks composed of tiny sensor nodes. Until now, most applications have treated the periodic small sized sensing data. Recently, the burst traffic with the transient and continuous nature is increasingly introduced due to the advent of wireless multimedia sensor networks. Therefore, the efficient communication protocol to support this trend is required.

In this paper, we propose a novel PIGAB(Packet Interval Gap based on Adaptive Backoff) protocol to efficiently transmit the burst data in multihop wireless sensor networks. The contention-based PIGAB protocol consists of the PIG(Packet Interval Gap) control algorithm in the source node and the MF(MAC-level Forwarding) algorithm in the relay node. The PIGAB is on basis of the newly proposed AB(Adaptive Backoff), CAB(Collision Avoidance Backoff), and UB(Uniform Backoff). These innovative algorithms and schemes can achieve the performance of network by adjusting the gap of every packet interval, recognizing the packet transmission of the hidden node. Through the simu-

· 본 연구는 건설교통부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과 Copyright@2008 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 제의 연구비지원(07국토정보C03)에 의해 수행되었습니다.

† 학생회원 : 광운대학교 전자통신공학과  
thno@adams.kw.ac.kr

†† 종신회원 : 광운대학교 전자통신공학과 교수  
kchung@kw.ac.kr

논문접수 : 2007년 6월 28일

심사완료 : 2008년 1월 18일

이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 정보통신 제35권 제3호(2008.6)

lations and experiments, we identify that the proposed PIGAB protocol considerably has the stable throughput and low latency in transmitting the burst data in multihop wireless sensor networks.

**Key words** : Multihop Wireless Sensor Networks, Burst Data, Backoff

## 1. 서론

최근 주변의 물리적인 현상을 인간에게 정보 형태로 제공하여 생활에 편리함을 줄 수 있는 무선 센서 네트워크(wireless sensor network)에 대한 연구가 활발히 진행되고 있다. 그리고 저비용 센서와 통신 네트워크의 기술 진보로 인해 무선 센서 네트워크의 실용화에 대한 기대도 커지고 있다. 이러한 무선 센서 네트워크는 센싱(sensing), 컴퓨팅, 그리고 무선 통신 능력을 갖춘 초소형의 센서 노드(node)로 구성되며, 이러한 세 가지 능력이 하나의 형태로 통합되어, 제한된 파워에 의존하여 서로 유기적으로 움직인다. 또한, 이러한 작은 센서 노드들은 서로 협력하여 멀티홉(multihop)으로 무선 통신을 하는 일반적인 주요 특징을 가지고 있다[1].

무선 센서 네트워크에 기반하여 자연 환경과 동식물의 서식지 모니터링, 건물 구조 모니터링, 공장관리, 홈네트워킹과 같은 다양한 응용 시나리오들이 개발되어 왔다. 이러한 응용들은 주기적인 형태의 비교적 단순한 센싱 데이터를 취급하는 것이 대부분이었다. 하지만 최근 다양한 형태의 센서 플랫폼(platform)들이 연구 개발됨에 따라 좀 더 요구사항이 복잡한 응용들이 새롭게 대두되고 있다. 이러한 응용들은 종전의 작은 센싱 데이터뿐만 아니라 일시적이고 연속적인 버스트(burst) 데이터를 취급할 수 있다. 따라서 이러한 형태의 응용을 효율적으로 지원하기 위한 전송 프로토콜에 대한 연구가 필요하다[2,3].

현재까지 멀티홉 무선 센서 네트워크 환경에서 버스트 데이터를 전송하기 위한 기법에 관한 연구는 거의 진행되지 않았다. 일부 논문에서 간략히 문제점을 거론하고는 있지만 구체적인 기법을 제시하지 못하고 있다[4,5]. 멀티홉 환경에서 데이터를 전송하기 위해서는 기본적으로 감춰진(hidden) 노드 문제를 해결해야 한다. 이러한 문제를 해결하기 위한 기존 방식으로 RTS/CTS(Request To Send/Clear To Send) 기법[6]이 가장 일반적이다. 하지만 패킷마다 RTS와 CTS를 남발하기 때문에 오버헤드(overhead)로 인해 많은 지연을 야기할 수 있다.

본 논문에서는 이러한 문제를 해결하기 위해 새롭게 제안하는 백오프(backoff) 기법을 기반으로 패킷의 전송 시기를 조절하는 PIGAB(Packet Interval Gap based on Adaptive Backoff) 프로토콜을 제안하였다. PIGAB

프로토콜은 근원지(source) 노드에서 동작하는 PIG(Packet Interval Gap) 제어 알고리즘과 릴레이(relay) 노드에서 동작하는 MF(MAC-level Forwarding) 알고리즘으로 구성된다. PIG 제어 알고리즘은 엿듣기 패킷(overhearing packet)에 담긴 백오프 값과 타임아웃(timeout)을 기반으로 다음 패킷의 전송 시기를 조절한다. MF 알고리즘은 수신된 패킷의 백오프 값을 기반으로 패킷의 전송 시기를 결정하여 다음 노드로 MAC 레벨에서 포워딩(forwarding)을 하는 방식이다. 이러한 두 알고리즘은 제안된 백오프 기법을 기반으로 동작하며, 멀티홉 무선 센서 네트워크 환경에서 버스트 데이터를 안정적이고 신속하게 전송하기 위한 목적을 가지고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 MAC 계층 프로토콜의 기존 연구 방향에 대해 기술하였고, 멀티홉 환경에서 버스트 데이터 전송에 따른 문제의 원인을 자세히 분석하였다. 3장에서는 2장에서 지적한 문제점을 해결하기 위해 본 논문에서 제안하는 PIGAB 프로토콜의 알고리즘에 대해 기술하였다. 4장에서는 제안한 프로토콜의 알고리즘을 시뮬레이션과 실제 실험을 통해 성능 검증을 하고, 마지막으로 5장에서 결론 및 향후 과제에 대해 기술하였다.

## 2. 관련 연구

본 장에서는 본 논문의 연구 방향으로 경쟁 기반 프로토콜을 선택한 이유와 무선 센서 네트워크 환경에서 버스트 데이터를 멀티홉으로 전송하는데 따른 문제점을 기술한다. 우선, 무선 센서 네트워크에서 MAC 계층 프로토콜 연구 방향에 대해 간략히 살펴보고, 그 중에서 경쟁 기반 프로토콜의 송수신 방식을 기술한다. 그리고 실제 실험을 통해 멀티홉 환경에서 버스트 데이터 전송상의 문제점을 분석한다.

### 2.1 MAC 계층 프로토콜

무선 센서 네트워크에서의 MAC 프로토콜 연구는 크게 세 가지 접근 방식으로 진행되고 있다. 우선, 경쟁 기반 프로토콜은 기존의 CSMA/CA(Carrier Sense Media Access/Collision Avoidance)[6] 방식을 바탕으로 랜덤하게 데이터를 전송하는 방식이다. 기존 CSMA/CA 방식의 심각한 에너지 낭비를 야기하는 유힬 듣기(idle listening) 문제는 시스템을 주기적으로 온-오프(on-off)하는 듀티 사이클(duty cycling) 기술로 해결하

였다. Polastre와 Culler에 의해 제안된 B-MAC[7]은 긴 프리엠블(preamble)을 사용하여 LPL(Low Power Listening)기법으로 동작하는 각 노드에서의 비동기적인 듀티 사이클을 에너지 효율적으로 지원하였다. 하지만 긴 프리엠블로 인한 엿듣기(overhearing) 문제의 에너지 비효율적인 문제를 해결하기 위해 X-MAC[8]이 제안되었다. X-MAC은 짧은 프리엠블에 목적지 주소를 담아 이러한 문제를 해결하였고, 빠른 ACK를 사용하여 계속되는 짧은 프리엠블을 멈추게 함으로써 B-MAC보다 환경에 동적으로 동작할 수 있다.

스케줄링(scheduling) 프로토콜은 기존의 TDMA(Time Division Media Access) 방식을 바탕으로 정해진 시간 순서대로 데이터를 전송하는 방식이다. 이러한 방식은 데이터 송수신상의 엄격한 시간 동기화가 필요하다는 단점을 갖지만, 충돌, 엿듣기, 유희 듣기 문제와 같은 경쟁 기반 프로토콜이 갖는 고질적인 문제에 완전히 자유롭다는 장점을 갖는다. 이러한 접근 방식의 대표적인 프로토콜로는 Hoese이 제안한 L-MAC[9]이 있다.

마지막으로, 무선 센서 네트워크를 위해 새롭게 등장한 접근 방식으로서, 경쟁 방식과 스케줄링 방식의 장점만을 취합한 슬롯(slot) 기반 프로토콜은 슬롯의 비활성 구간에서 휴지(sleep) 상태로 존재하고, 활성 구간을 이용하여 데이터를 전송하는 방식이다. 이러한 방식은 TDMA 방식에 비해 엄격하진 않지만, 데이터 송수신상의 시간 동기화에 대한 오버헤드를 갖는다. Ye에 의해 제안된 S-MAC[4]은 각 노드에서의 동기적인 듀티 사이클을 기반으로 고정된 활성 구간내에 데이터를 전송

한다. 하지만 S-MAC은 고정된 활성 구간으로 인해 트래픽(traffic) 변화에 동적으로 대응하지 못하는 문제를 갖는다. 이러한 문제를 해결하기 위해 활성 구간을 적응적으로 변화시키는 T-MAC[10]이 제안되었다.

MAC 프로토콜에 관한 지금까지의 연구는 주로 에너지 효율성에 편중되어 왔으며, 주기적인 작은 크기의 센싱 데이터를 에너지 효율적으로 전송하기 위한 기법들을 연구하는데 총력을 기울여 왔다. 하지만 응용에 따라 일시적으로 발생 가능하며 연속적인 패킷으로 이루어진 버스트 데이터를 멀티-홉으로 전송하기 위한 구체적인 연구가 진행되지 않았다. 그리고 MAC 프로토콜 연구에 있어 다른 접근 방식과 비교하여 경쟁 기반 프로토콜은 여러 가지 장점을 가지고 있다. 상대적으로 알고리즘이 간단하고, ad-hoc 성향에 적합하며, 이동성과 트래픽(traffic) 변화에 유동적이다. 또한 확장성이 뛰어나며, 시간 동기화에 대한 오버헤드가 없다는 것이다[11]. 따라서 본 논문에서는 이러한 이유 때문에 경쟁 기반 프로토콜에 버스트 데이터에 대한 전송 기법을 추가하는 방향으로 연구를 진행하였다.

2.2 경쟁 기반 프로토콜의 송수신 방식

그림 1과 2는 실제 사용되고 있는 경쟁 기반 프로토콜의 송수신 방식을 나타낸 것이다. 기존의 CSMA/CA[6] 특징을 따르고 있으며, 감춰진 노드 문제로 인한 충돌을 완화하기 위한 RTS/CTS(Request To Send/ Clear To Send) 기법과 MAC 레벨의 신뢰성을 위한 ACK 기법을 선택적으로 추가할 수 있는 구조를 가지고 있다.

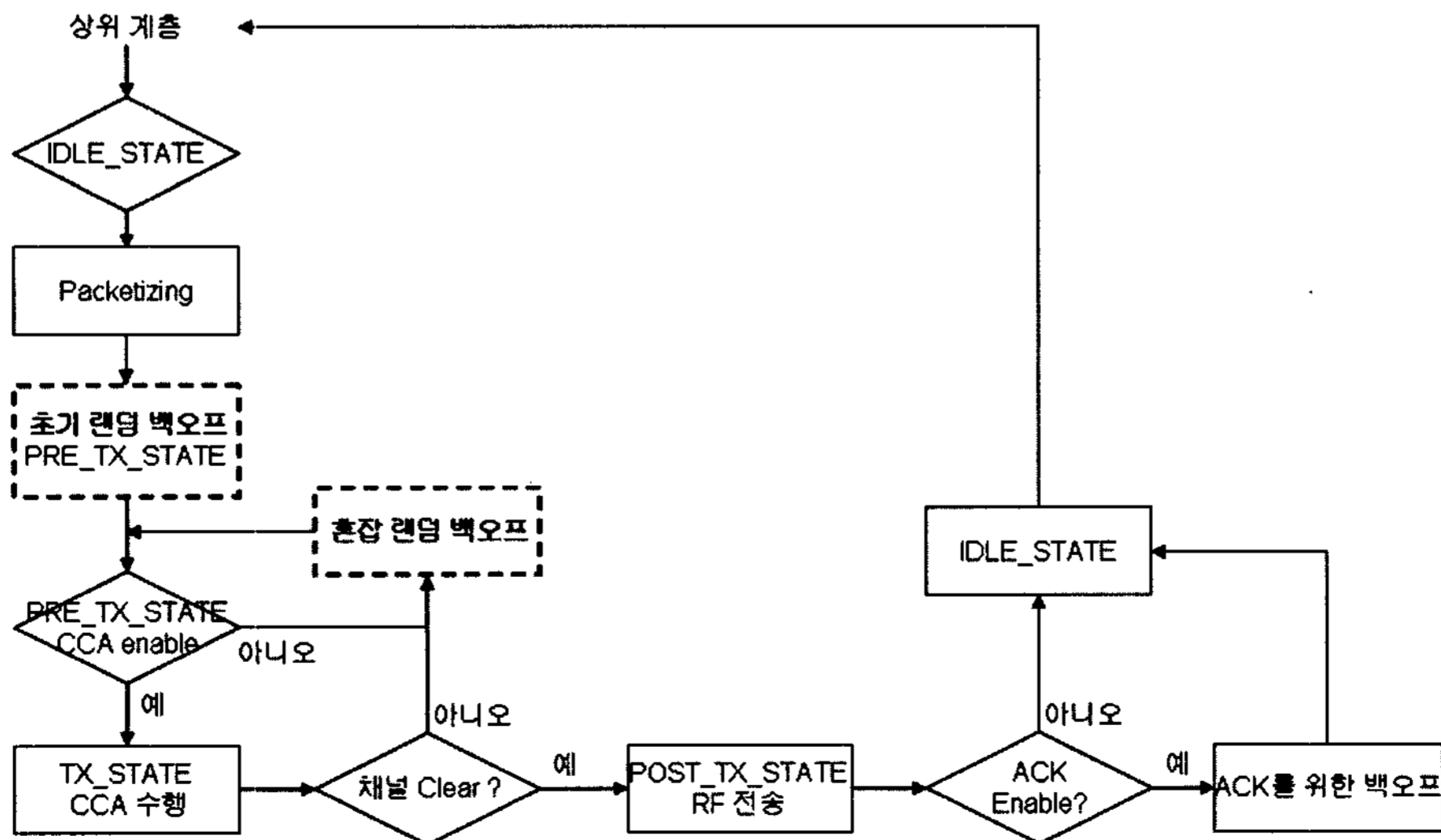


그림 1 경쟁 기반 프로토콜의 송신 방식

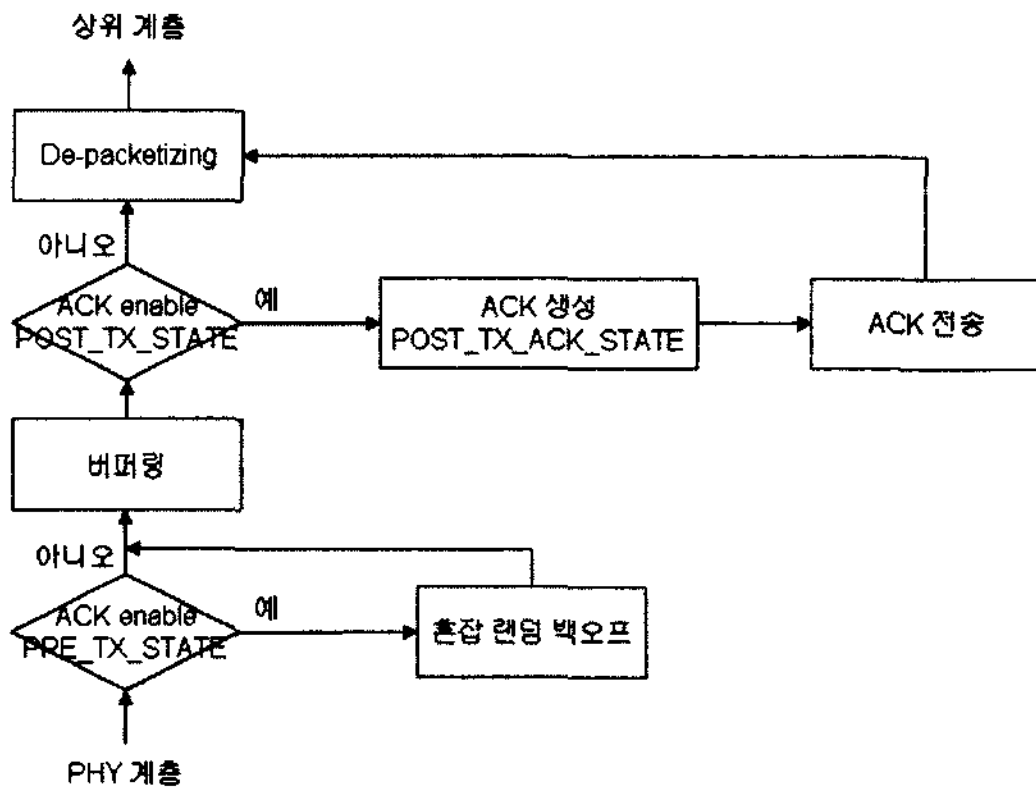


그림 2 경쟁 기반 프로토콜의 수신 방식

데이터 전송의 경쟁을 최소화하고 충돌을 회피하기 위해 최대 백오프(backoff) 범위에서 랜덤하게 선택된 값으로 백오프를 하는 초기 백오프 기법과 CCA(Clear Channel Assessment) 기반의 혼잡 백오프 기법을 사용하는 것을 그림 1에서 확인할 수 있다. 이러한 기법을 통해 상위에서 받은 패킷을 실제 전송하게 된다. 초기 백오프 기법은 선택적으로 구현될 수 있지만 경쟁을 최소화하기 위해 일반적으로 사용된다. 그러나 무선 센서 네트워크 환경에서 연속적인 패킷의 버스트 데이터를 멀티-홉으로 전송하기에 기존의 백오프 기법은 바람직하지 않다. 그 이유에 대해서 이후의 절에서 원인 분석을 통해 확인 할 것이다. 한편, 경쟁 기반 프로토콜에서의 데이터 수신은 선택적으로 ACK에 대해 적절히 처리하는 구조를 가지고 있으며, 상위 계층으로 패킷 수신을 알리는 구조로 되어 있다.

### 2.3 문제 분석

본 절에서는 멀티홉 무선 센서 네트워크 환경에서 버스트 데이터 전송에 따른 문제점을 분석하기 위해 실험

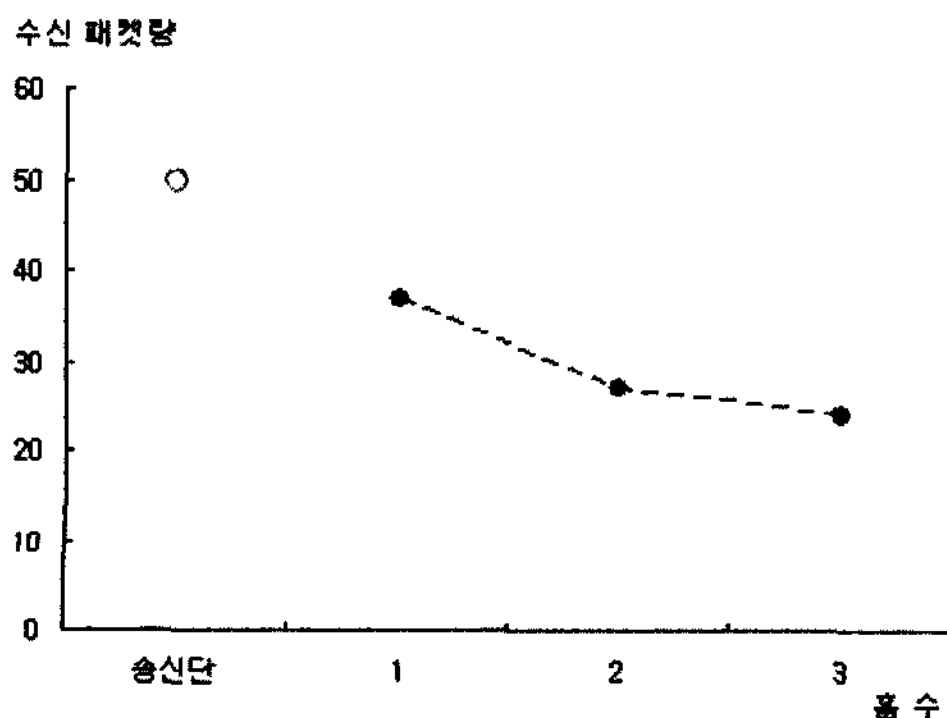


그림 4 MAC 레벨 throughput

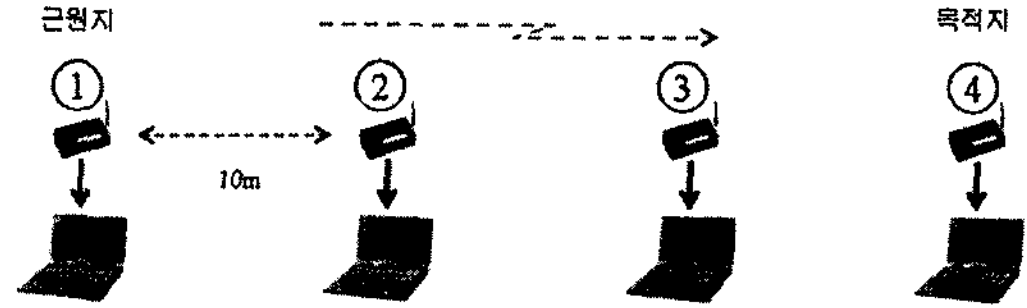


그림 3 실험 환경

을 수행하였다.

그림 3은 실험 환경을 나타낸 것이며, 센서 플랫폼은 캘리포니아 버클리 대학에서 개발된 MICAZ[12]을 사용하여 실험을 수행하였다. 다음과 같은 환경에 CBR (Constant Bit Rate) 패턴으로 연속적으로 50개의 패킷을 주입하고, 패킷 스니퍼(packet sniffer)를 이용하여 패킷의 흐름을 분석하였다. 그에 따른 결과로써, 그림 4는 일정한 패킷 전송량에 따른 MAC 레벨 수신률을 보여주며, 홉에 따라 수신률이 감소되는 것을 확인할 수 있다. 무선상의 손실이 없다고 가정할 때, 이러한 성능 저하는 대부분 멀티홉 환경에서의 감춰진 노드 문제로 인해 홉에 따라 패킷이 손실되었기 때문으로 분석할 수 있다. 그림 5는 그 결과를 나타낸 것이다. 근원지 노드의 전송 패킷은 감춰진 노드 3에 대한 간섭으로 인해 1홉에서 대략 30%의 패킷이 손실되었음을 확인할 수 있다. 또한 2홉에서 25%, 3홉에서 12%의 패킷이 손실되었음을 확인할 수 있다.

본 논문에서는 실험 결과를 기반으로 문제를 면밀히 분석하고, 문제 해결을 위한 적절한 방법을 찾고자 한다. 표 1은 원인 분석을 위해 정의한 파라미터를 나타낸 것이다.

그림 6은 정의된 파라미터를 적용한 분석 모델을 나타낸 것이며, S는 근원지 노드, R은 릴레이(relay) 노드, 그리고 H는 감춰진 노드로 정의하였다. 노드 S에서  $i-1$ 번째 패킷은 응용 계층에서의  $PI_{APP}$  후에 하위 계

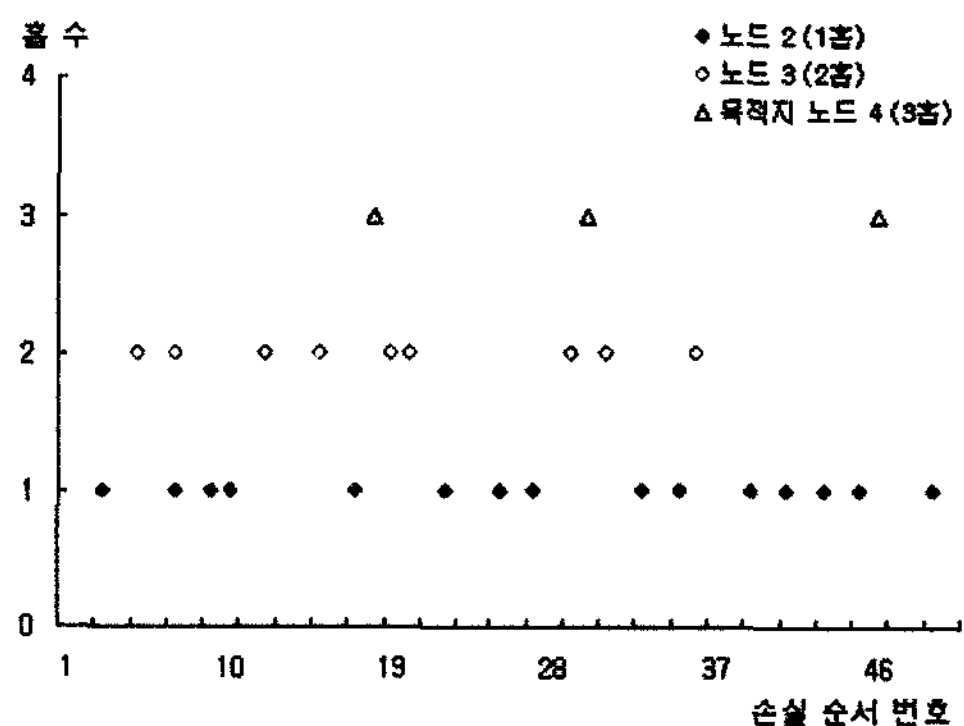


그림 5 홉에 따른 손실 패킷

표 1 파라미터 정의

파라미터	설명
$\alpha$ (알파)	최대 백오프(Back-off) 범위
$\beta_{ID,(순서번호)} = rand(0, \alpha)$ (베타)	랜덤 백오프 값
$TP_{ID,(순서번호)}$	백오프 후 패킷 전송
$PI_{APP}$	응용 계층에서의 패킷 인터발
$\delta$ (델타)	충돌 범위

충으로 전송되고, MAC 계층에서의  $\beta_{S,(i-1)}$  후에 노드 R로 실제 전송된다. 이렇게 노드 R로 전송된  $i-1$ 번째 패킷은  $\beta_{R,(i-1)}$  후에 노드 H로 전송되며,  $\beta_{H,(i-1)}$  후에 노드 H에서 다음 노드로 전송된다. 이러한 방식으로  $i-1$ 번째 패킷이 최종 목적지로 전송된다. 한편, 일정한  $PI_{APP}$ 와 랜덤한  $\beta_{S,(i)}$  후에  $i$ 번째 패킷이 전송되며, 위에 설명한 동일한 방식으로 최종 목적지로 전송된다. 하지만 노드 S의  $i$ 번째 패킷은 노드 H의  $i-1$ 번째 엿듣기 패킷으로 인해 간섭과 충돌 문제가 발생할 수 있다.

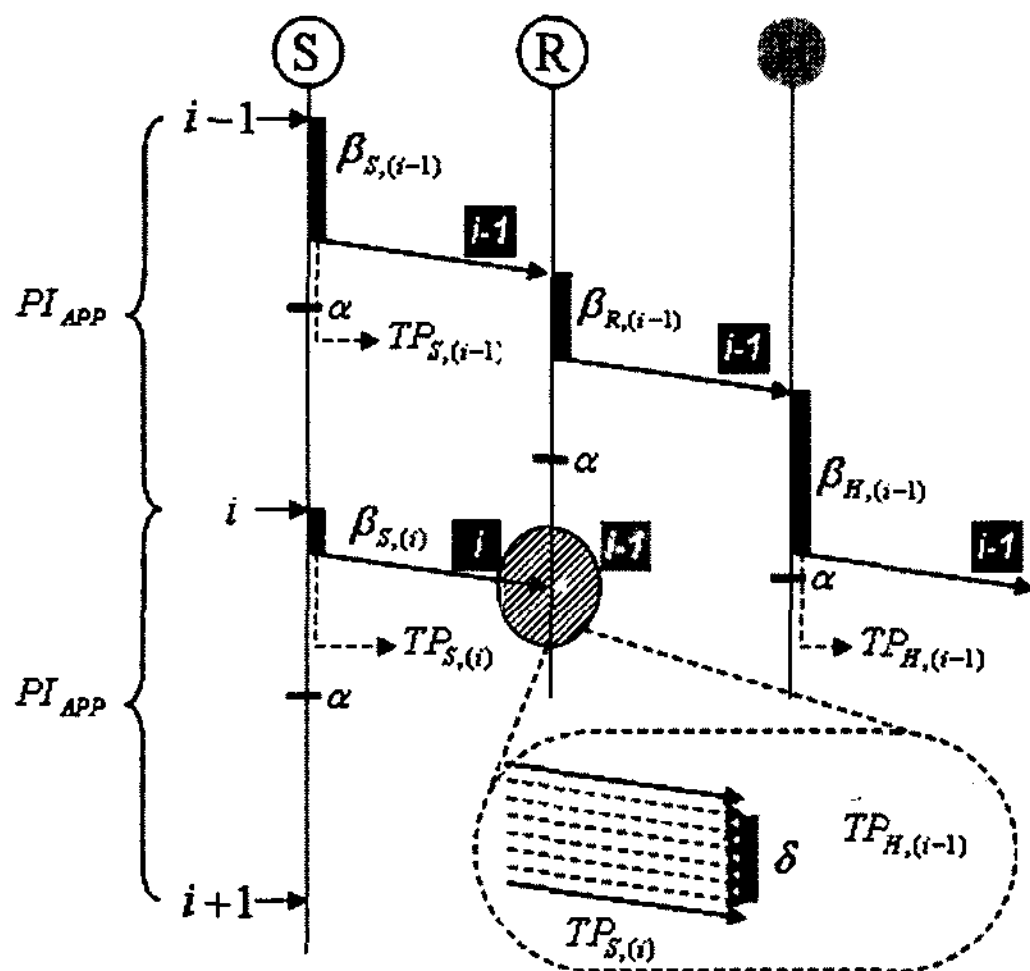


그림 6 분석 모델

이러한 부정적인 영향을 정량적으로 분석하기 위해 Matlab[13]을 사용하여 충돌 확률을 측정하였다. 노드 H의  $TP_{H,(i-1)}$ 을 기준으로 노드 S에서 랜덤하게 전송되는  $TP_{S,(i)}$ 와의 충돌을 확인하였다. 그림 6에서  $\delta$ 은 패킷의 송수신을 위해 물리적으로 발생하는 TX/RX (Transmit/Receive)의 교환 시간과 패킷을 수신하여 처리하는 시간을 고려한 파라미터이며, 시스템에 따라 가변적일 수 있다. 식 (1)은  $\delta$ 을 적용하여 충돌 확률 계산에 사용된 충돌 확률 범위를 정의한 것이며, 이러한 범위 내에서  $TP_{S,(i)}$ 가 수신되면 충돌이라고 판단하여 충돌 확률을 계산하였다.

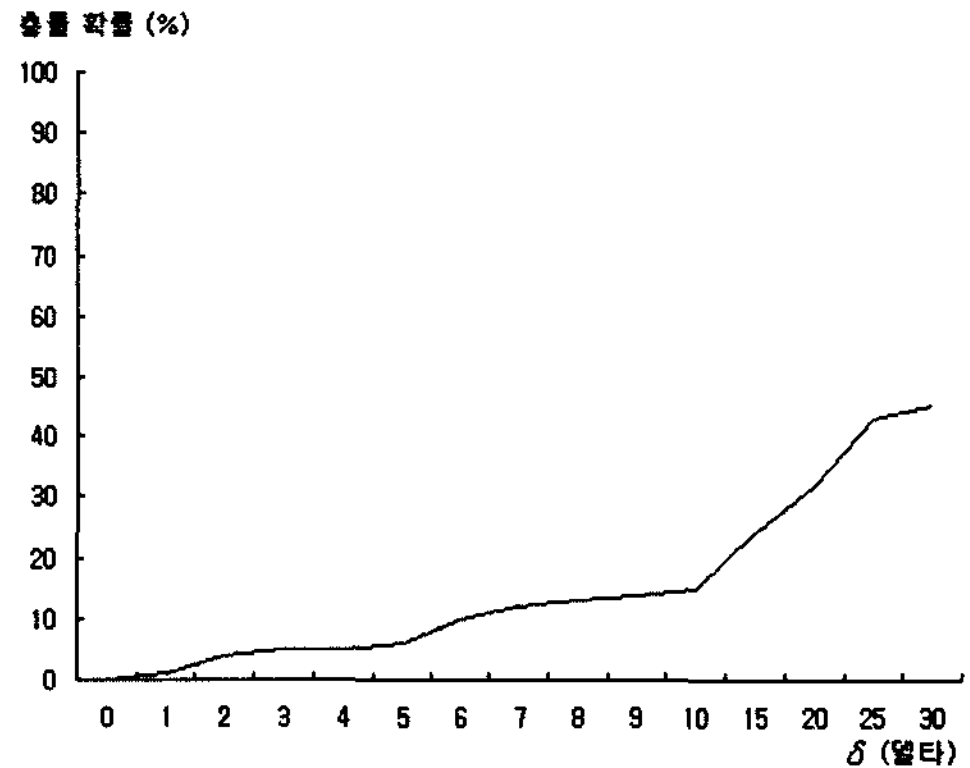


그림 7 충돌 확률

$$TP_{S,(i)} - \delta < TP_{H,(i-1)} \leq TP_{S,(i)} \quad (1)$$

그림 7은 식 (1)을 적용하여 노드 S의  $i$ 번째 패킷과 노드 H의  $i-1$ 번째 엿듣기 패킷과의 충돌 확률을 나타낸 것이다.  $\delta$ 범위의 변화에 따라 충돌 확률이 증가되는 것을 확인할 수 있다.

이러한 충돌 문제에 대한 원인은 근원지 노드에서의 패킷 전송 시기와 감춰진 노드에서의 이전 패킷의 전송 시기를 바탕으로 파악할 수 있다. 식 (2)에서  $TP_{S,(i)}$ 은 근원지 노드에서의 패킷 전송 시기를 의미하며, 응용 계층에서의 일정한  $PI_{APP}$ 와 MAC 계층에서의 일정한 최대 백오프 범위에서 랜덤하게 선택된  $\beta_{S,(i)}$ 을 더한 값으로 나타낼 수 있다. 그리고 식 (3)에서  $TP_{H,(i-1)}$ 은 감춰진 노드에서의 이전 패킷의 전송 시기를 의미하며, 마찬가지로 MAC 계층에서의 일정한 최대 백오프 범위에서 랜덤하게 선택된  $\beta_{H,(i-1)}$  값으로 나타낼 수 있다.

$$TP_{S,(i)} = PI_{APP} + \beta_{S,(i)} \quad (2)$$

$$TP_{H,(i-1)} = \beta_{H,(i-1)} \quad (3)$$

이러한 두 관계식에서 충돌에 영향을 주는 요소는 근원지 노드의 응용 계층에서 패킷이 일정하게 전송되는 시기(interval)와 각 노드에서의 일정한 최대 백오프 범위에서 랜덤하게 선택되는 백오프 값이다. 즉, 근원지 노드는 멀티홉 감춰진 노드의 백오프 범위를 인지하지 못하고, 다음 패킷을 전송함에 따라 간섭이나 충돌로 인한 패킷 손실을 야기한다. 결국, 이것은 시스템 성능을 저하시킨다. 따라서 멀티홉 감춰진 노드로 인한 충돌 문제를 완화하기 위해 근원지 노드에서는 감춰진 노드의 백오프 범위를 인지하여 다음 패킷의 전송 시기를 조절하는 기법이 필요하다.

무선 센서 네트워크에서 일시적으로 발생 가능하며 연속적인 패킷으로 이루어진 버스트 데이터를 멀티홉으

로 전송하기 위한 구체적인 연구가 지금까지 진행되지 않았다. 또한 멀티홉 감춰진 노드 문제를 해결하기 위한 기존의 CSMA/CA 기반의 RTS/CTS 기법은 데이터 패킷마다 제어 패킷의 오버헤드로 인해 상당한 지연을 야기하기[4,5] 때문에 바람직한 해결 방법이 아니다. 따라서 본 논문은 정량적인 원인 분석을 통해 멀티홉 감춰진 노드로 인한 충돌 문제를 해결하여 패킷 손실에 대한 복구 기법 없이 패킷 전송에 따른 안정적인 수신률과 낮은 지연을 갖도록 버스트 데이터를 멀티홉으로 전송하는 것을 목적으로 한다.

### 3. PIGAB 프로토콜 알고리즘

본 장에서는 2장에서 지적한 문제점을 해결하기 위해 새롭게 제안하는 PIGAB(Packet Interval Gap based on Adaptive Backoff) 프로토콜의 알고리즘에 대해 기술한다. 그리고 제안한 PIGAB 프로토콜을 지원하기 위해 새롭게 제안된 AB(Adaptive Backoff), CAB(Collision Avoidance Backoff), 그리고 UB(Uniform Backoff) 기법에 대해 기술한다. PIGAB 프로토콜은 무선 센서 네트워크 환경에서 멀티홉 감춰진 노드와의 충돌 문제를 상황에 동적으로 적용되는 제안된 백오프 기법으로 해결함으로써 버스트 데이터를 안정적으로 신속하게 전송할 수 있다.

#### 3.1 PIGAB 프로토콜 구조

무선 센서 네트워크에서 일시적으로 발생 가능하며 연속적인 패킷으로 이루어진 버스트 데이터를 멀티홉으로 전송하기 위한 구체적인 연구가 지금까지 진행되지 않았다. 또한 멀티홉 감춰진 노드 문제를 해결하기 위한

기존의 접근 방식은 제어 패킷의 상당한 오버헤드로 인해 바람직한 해결 방법이 아니다. 그리고 근원지 노드가 감춰진 노드의 백오프 범위를 인지하지 못하고 다음 패킷을 전송함으로써, 충돌로 인해 성능을 저하시키는 것을 정량적인 원인 분석을 통해 확인하였다.

본 논문에서는 이러한 문제를 해결하기 위해 상황에 동적으로 적용되는 백오프 기법을 기반으로 버스트 데이터를 멀티홉으로 전송하는 PIGAB 프로토콜을 제안하였다. PIGAB 프로토콜은 근원지 노드에서 감춰진 노드의 백오프 범위를 인지하여 다음 패킷의 전송 시기를 조절하게 된다.

그림 8은 PIGAB 프로토콜 구조를 나타낸 것이다. PIGAB 프로토콜은 기본적으로 경쟁 기반 프로토콜을 기반으로 하며, 송신단에서 동작하는 PIG 제어 알고리즘과 수신단에서 동작하는 MF 알고리즘으로 구성된다. PIG 제어 알고리즘은 수신단의 엿듣기 패킷에 담긴 백오프 값과 타임아웃(timeout)을 기반으로 각 패킷간의 전송 시기를 조절하게 된다. 또한, MF 알고리즘은 수신된 백오프 값을 기반으로 패킷을 목적지로 빠르게 전송하게 된다.

이후 3.2절에서는 패킷간의 전송 시기를 동적으로 조절하기 위해 제안된 세 가지 백오프 기법에 대해 설명하고, 3.3과 3.4절에서는 PIG 제어 알고리즘과 MF 알고리즘에 대해 상세하게 기술하였다.

#### 3.2 제안하는 백오프 기법

PIGAB 프로토콜의 패킷 전송 시기를 상황에 따라 동적으로 조절하기 위해 AB, CAB, 그리고 UB 기법을 새롭게 제안한다. 이러한 백오프 기법들은 임계값(thres-

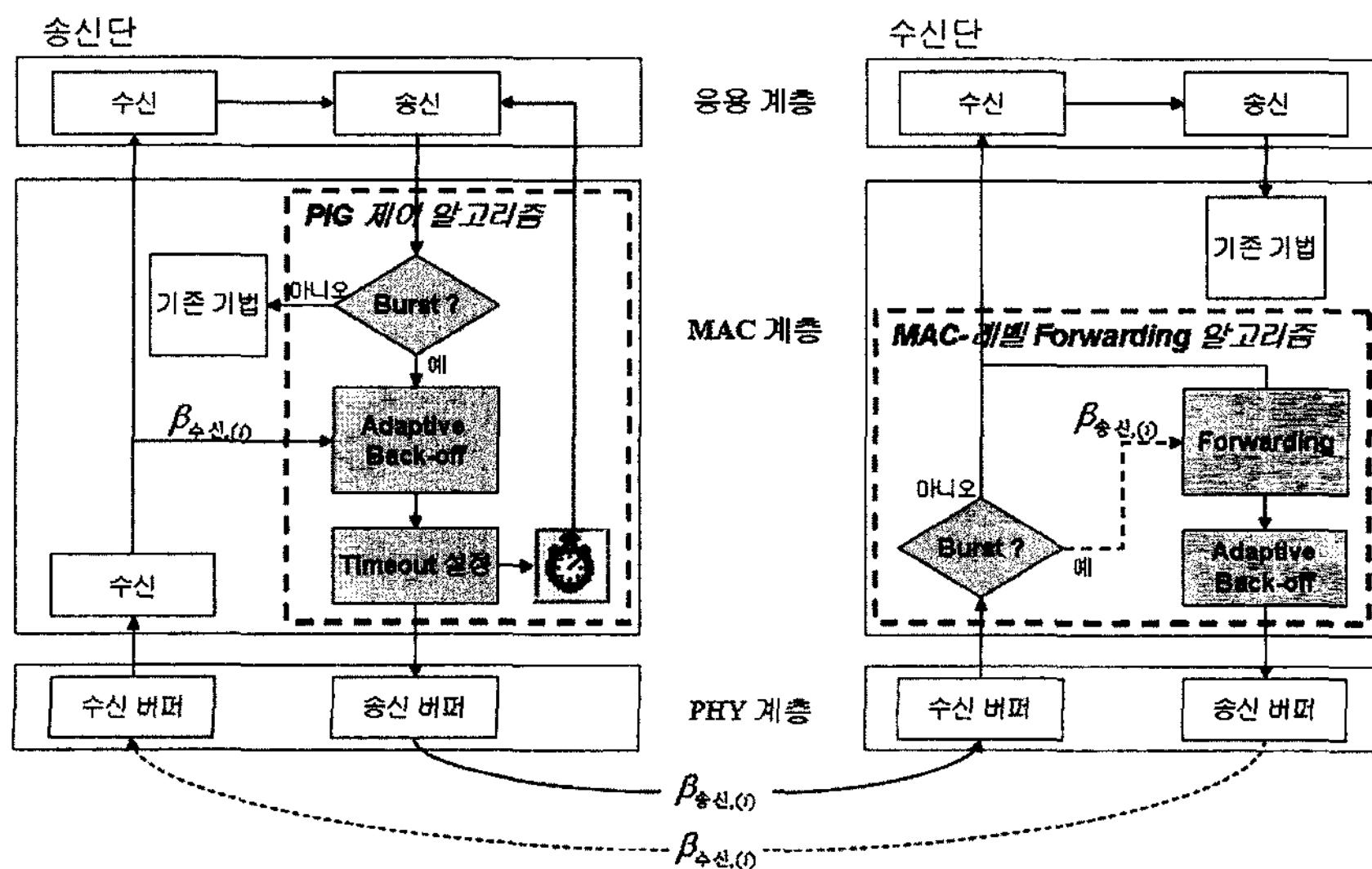


그림 8 PIGAB 프로토콜 구조

hold)를 기준으로 동작하게 된다. AB는 모든 노드에서 사용되는 기본적인 방식이며, CAB은 충돌 회피를 위해 근원지 노드에서 사용되고, UB는 릴레이(relay) 노드에서 사용된다.

3.2.1 AB(Adaptive Backoff) 기법

AB는 감춰진 노드와의 충돌을 회피하고 패킷을 신속하게 전송하기 위한 목적으로 설계되었다. 그림 9는 AB의 동작 방식을 나타낸 것이다. 최초 근원지 노드에서의  $\beta_{\text{근원지},(i)}$ 는 최대 백오프 범위( $\alpha$ )에서 랜덤하게 선택되고, 다음 노드에서의  $\beta_{\text{릴레이X},(i)}$ 는 수신된  $\beta_{\text{근원지},(i)}$ 를 기반으로 그 범위에서 랜덤하게 선택된다. 이러한 방식으로 AB는 이전 노드의 백오프 값을 기반으로 최대 백오프의 범위를 줄여감에 따라 충돌을 회피하고, 패킷을 신속하게 전송할 수 있다.

하지만 이전 노드의 백오프 값을 기반으로 백오프의 범위를 줄여가는 AB 기법은 랜덤 범위가 감소하기 때문에 충돌 확률이 증가하는 문제를 가지고 있다. 이러한 문제를 보완하기 위해  $Thresh_{CD}$ (충돌 감지 임계값)를 정의하여 그 이하로 랜덤 범위가 감소하지 않도록 설정하였다. 따라서  $Thresh_{CD}$ 를 기준으로 그 이상일 때 기본적인 AB로 동작하게 되고, 그 이하일 때 근원지 노드에서는 CAB으로 릴레이 노드에서는 UB로 동작하게 된다.

$$\beta_{\text{근원지},(i)} = rand(0, \alpha)$$

$$\beta_{\text{릴레이X},(i)} = rand(0, \beta_{\text{근원지},(i)})$$

$$\beta_{\text{릴레이Y},(i)} = rand(0, \beta_{\text{릴레이X},(i)})$$

$$\beta_{\text{릴레이Z},(i)} = rand(0, \beta_{\text{릴레이Y},(i)})$$

그림 9 AB 기법

3.2.2 CAB(Collision Avoidance Backoff) 기법

CAB은 릴레이 노드와의 충돌 회피를 위해 근원지 노드에서 사용되는 백오프 기법이다. 그림 10은 CAB의 동작 방식을 나타낸 것이다. 초기 패킷은 각 노드에서의 백오프 값이  $Thresh_{CD}$  이상이 되어 AB 기법으로 전송되는 것을 확인할 수 있다. 다음 패킷들은 릴레이X에서 수신된 백오프 값을  $Thresh_{CD}$ 와 비교하여 AB와 CAB 중 어떤 기법을 사용할 것인지 판단 후에 상황에 따라 동적으로 패킷을 전송하게 된다. 만일 이전 노드가  $Thresh_{CD}$ 에 걸리지 않았을 경우, AB로 다음 패킷을 전송한다.

반면에, 이전 노드가  $Thresh_{CD}$ 에 걸릴 경우, AB를 그대로 적용할 수 있으나, 백오프 범위가 너무 줄어들게 된다. 또한 기존의 백오프 기법을 사용하게 되면, 백오프 값이 랜덤하게 선택되기 때문에  $Thresh_{CD}$  이하로 선택될 확률이 있고, 감춰진 노드의 엿듣기 패킷과의 충돌 문제가 발생하게 된다. 이러한 문제를 해결하기 위해  $Thresh_{CA}$ (충돌 회피 임계값)를 정의하여 백오프 값을 그 이상으로 선택하여 패킷을 전송하게 된다. 따라서 CAB은 이전 노드가  $Thresh_{CD}$ 에 걸릴 경우  $Thresh_{CA}$  이상의 백오프 후 패킷을 전송함으로써 이전 패킷과의 충돌을 미연에 방지할 수 있다.

3.2.3 UB(Uniform Backoff) 기법

UB 기법은 패킷 릴레이 도중에 충돌을 회피하고 패킷을 신속하게 전송하기 위해 릴레이 노드에서 사용되는 백오프 기법이다. 그림 10은 UB의 동작 방식을 나타낸 것이다. 모든 노드는 패킷을 수신할 경우 이전 노드의 백오프 값이  $Thresh_{CD}$ 에 걸렸는지 항상 확인한다. 만일  $Thresh_{CD}$ 에 걸릴 경우 근원지 노드는 CAB를 사용하지만, 릴레이 노드는 UB를 사용한다. UB를 통해 모든 노드들은 주변 노드의 백오프 상황을 인지할 수

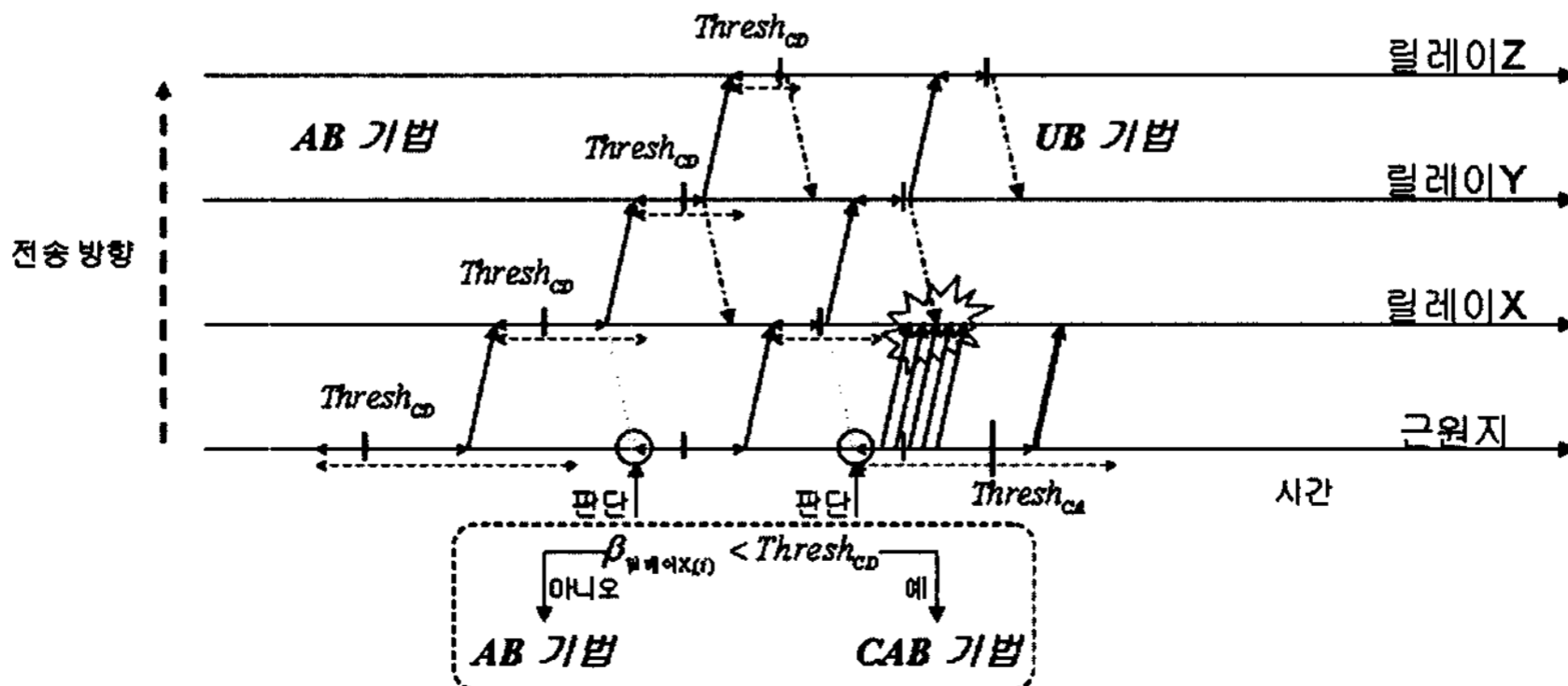


그림 10 CAB과 UB 기법

있게 된다. 그리고 UB는 기본적인 랜덤 기법을 대신하여 작고 일정한 백오프 범위로 패킷을 전송하게 된다. 따라서 UB는 이러한 방식을 통해 충돌을 회피하고 패킷을 신속하게 전송할 수 있다.

3.3 PIG(Packet Interval Gap) 제어 알고리즘

본 절에서는 수신단 노드의 백오프 값과 타임아웃을 기반으로 패킷의 전송 시기를 조절하는 근원지 노드에서 동작하는 PIG 제어 알고리즘을 상세히 설명한다. 그리고 실제 센서 플랫폼에 구현된 PIG 제어 알고리즘을 의사 코드(pseudo code)로 정리한다.

PIG 제어 알고리즘은 감춰진 노드와의 충돌을 줄여 전송률을 안정화시키고, 제안된 백오프 기법을 기반으로 지연을 최소화하려는 목적으로 설계되었다. PIG 제어 알고리즘은 수신단 노드의 엿듣기 패킷에 담긴 백오프 값과 패킷마다 계산되는 타임아웃을 기반으로 각 패킷의 전송 시기를 조절하는 근원지 노드에서 동작하는 알고리즘이다.

다음 패킷의 전송 시기를 결정하는 절차는 패킷 전송과 함께 그 패킷의 타임아웃을 계산하고, 계산된 타임아웃 동안 수신단 노드의 엿듣기 패킷을 기다리게 된다. 패킷의 전송 시기를 조절하는데 이용되는 타임아웃은 레디오(radio) 신호의 불규칙성으로 인해 엿듣기 패킷이 비신뢰적으로 수신되는 것을 방지하기 위해 고려하였다. 그림 11은 타임아웃이 계산되는 것을 나타낸 것이다.  $i$  번째 패킷의 타임아웃은  $\beta_{\text{근원지},(i)}$ 와 패킷 크기를 최대 백오프 범위와 비교하여 결정된다.  $\beta_{\text{근원지},(i)}$ 가 다르기 때문에 패킷마다 가변적인 타임아웃을 갖게 되며, 이렇게 계산된 타임아웃 동안 수신단 노드의 엿듣기 패킷을 기다리게 되고, 수신되면 즉시 상위 계층으로 알려 AB

$$\text{if}(\alpha < \beta_{\text{근원지},(i)} + \text{PACKET\_SIZE})$$

$$T_{\text{근원지},\text{타임아웃}(i)} = \beta_{\text{근원지},(i)} + \text{PACKET\_SIZE}$$

else

$$T_{\text{근원지},\text{타임아웃}(i)} = \alpha$$

그림 11 타임아웃 계산

혹은 CAB 기법으로 다음 패킷을 전송하게 된다. 그렇지 않은 경우, 타임아웃 후 AB 혹은 CAB으로 패킷을 전송하게 된다.

그림 12는 PIG 제어 알고리즘의 구조를 나타낸 것이다. PIG 제어 알고리즘은 크게  $i-1$ 번째 패킷 전송에 사용될 백오프를 결정하는 부분, 그 패킷의 타임아웃을 계산하는 부분, 그리고  $i$ 번째 패킷에 사용될 백오프를 결정하는 부분으로 구성되어 있다. PIG 제어 알고리즘에 사용된 변수는 WhichBackoff와 Adaptive\_Beta이며, WhichBackoff는 패킷 전송에 사용될 백오프를 결정하는데 사용되며, 초기에 COLLISION\_AVOID로 정의되어 있다. 수신된  $\beta_{\text{릴레이},(i)}$ 가  $\text{Thresh}_{CD}$ 에 걸렸는지 판단하여 WhichBackoff가 상황에 따라 동적으로 변화되며, 그 수신된 백오프를 Adaptive\_Beta에 저장하여 AB로 다음 패킷을 전송하는데 사용한다.

최초에 상위 계층에서 받은 데이터의 유형을 확인하여 단일 형태의 센싱 데이터일 경우 기존 방식으로 패킷을 전송하고, 연속적인 형태의 버스트 데이터일 경우 PIG 제어 알고리즘으로 패킷을 전송하게 된다. 첫 번째 버스트 패킷은  $\beta_{\text{근원지},(1)}$ 가 너무 작게 선택되는 것을 피하기 위해  $\text{Thresh}_{CA}$ 와 비교하여,  $\beta_{\text{근원지},(1)}$ 가 큰 경

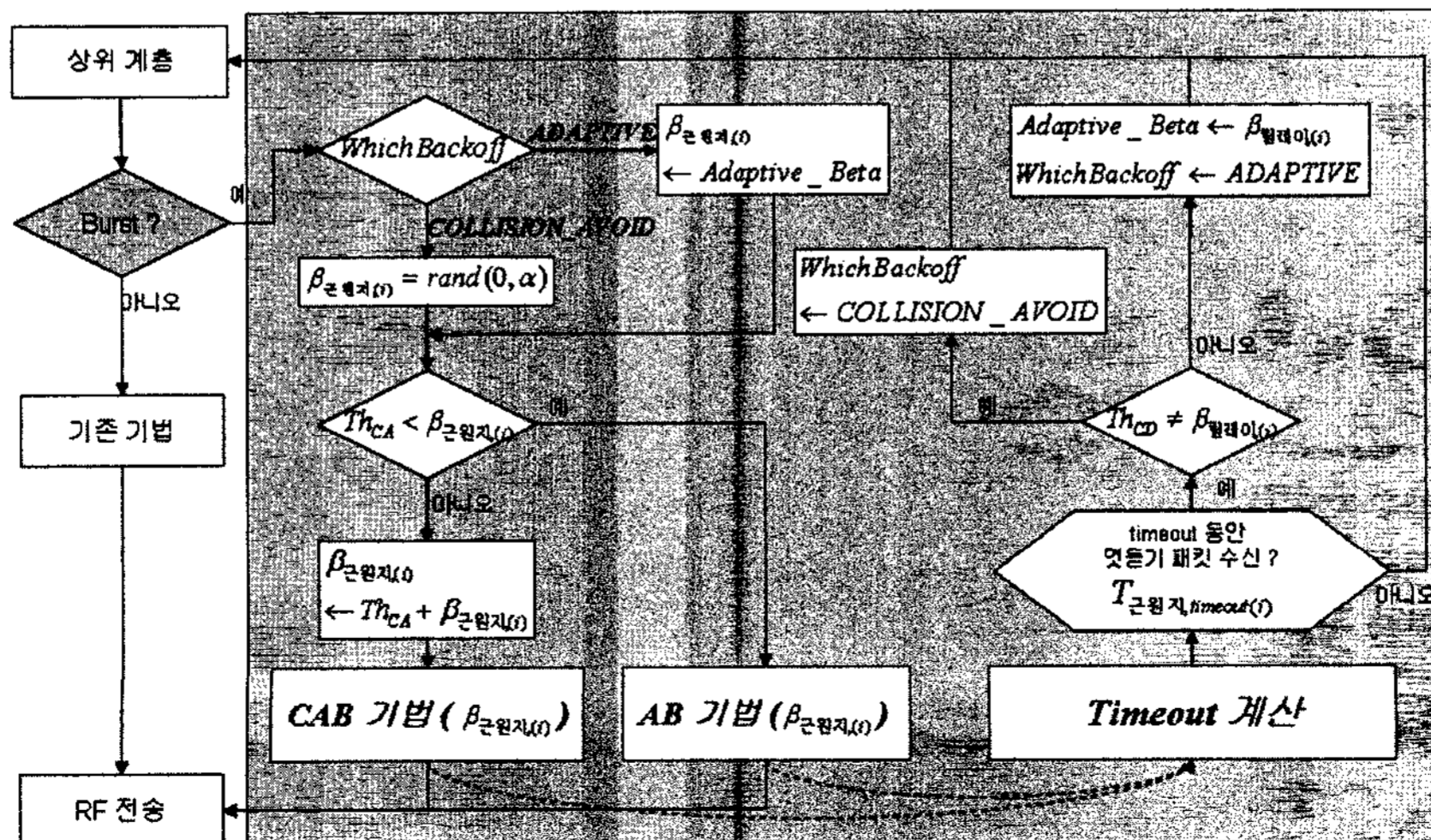


그림 12 PIG 제어 알고리즘



우 AB로 전송된다. 그렇지 않은 경우 식 (4)를 이용하여  $\beta_{\text{근원지},(1)}$ 를 갱신한 후 CAB으로 첫 번째 버스트 패킷을 전송하게 된다.

$$\beta_{\text{근원지},(1)} = \text{Thresh}_{CA} + \beta_{\text{근원지},(1)} \quad (4)$$

첫 번째 패킷의 전송과 함께 그 패킷의 타임아웃은 위에서 설명한 방식으로 계산된다. 이렇게 계산된 타임아웃 동안 수신단의 엿듣기 패킷을 기다리고, 패킷이 수신되지 않으면 타임아웃 후 응용 계층으로 즉시 신호를 주어 그 다음 패킷이 하위 계층으로 전달될 수 있도록 한다. 따라서 두 번째 패킷은 첫 번째 패킷과 동일한 방식으로 전송된다. 하지만 패킷이 수신되면 패킷에 담긴  $\beta_{\text{릴레이},(1)}$ 을 추출하여  $\text{Thresh}_{CD}$ 에 걸렸는지 판단한 후, 걸렸을 경우 다음 패킷이 CAB로 전송될 수 있도록 유도하고,  $\text{Thresh}_{CD}$ 에 걸리지 않았을 경우 다음 패킷이 AB로 전송될 수 있도록 한다.

지금까지 설명한 PIG 제어 알고리즘을 의사 코드로 정리하였다. 이러한 의사 코드를 바탕으로 MICAZ의 MAC 계층에 제안하는 알고리즘을 구현하였다. PIG 제어 알고리즘은 수신단 노드의 엿듣기 패킷에 담긴 백오프 값과 패킷마다 계산되는 타임아웃을 기반으로 각 패킷의 전송 시기를 조절하게 된다. 따라서 PIG 제어 알고리즘은 무선 센서 네트워크 환경에서 감춰진 노드와의 충돌을 해결할 수 있기 때문에 버스트 데이터를 멀티홉으로 안정적이고 신속하게 전송할 수 있다.

### 3.4 MF(MAC-level Forwarding) 알고리즘

본 절에서는 수신된 패킷에 담긴 백오프 값을 기반으로 패킷을 다음 목적지로 전송하는 수신단 노드에서의 MF 알고리즘을 상세히 설명한다. 그리고 실제 센서 플랫폼에 구현된 MF 알고리즘을 의사 코드(pseudo code)로 정리한다. MF 알고리즘은 이웃 노드의 백오프를 인지하여 패킷 충돌 없이 MAC 레벨에서 패킷을 신속하게 전송하려는 목적으로 설계되었다. MF 알고리즘은 수신된 패킷에 담긴 백오프 값을 기반으로 패킷을 전송하는 알고리즘이다.

그림 14는 MF 알고리즘의 전체 구조를 나타낸 것이다. MF 알고리즘은 크게 MAC 레벨에서 다음 노드로 패킷을 포워딩(forwarding)할 것인지를 판단하는 부분과 수신된 백오프를 기반으로 AB 혹은 UB로 패킷을 전송하는 부분으로 구성되어 있다. 패킷 전송에 사용될 백오프를 결정하기 위해 *WhichBackoff*를 정의하였으며, 초기에 ADAPTIVE로 설정되어 있다.

최초에 하위 계층에서 받은 패킷이 어떤 유형인지 확인하여 단일 형태의 센싱 데이터일 경우 기존 방식으로 처리하고, 연속적인 형태의 버스트 데이터일 경우 MF 알고리즘이 동작한다. 수신된 버스트 패킷의  $\beta_{ID,(i)}$ 와

Input: Packet from the Upper Layer

Output: Backoff Scheme and Packet Transmission

Whois=Source,  $\text{Thresh}_{CA} = \alpha / 2$ , WhichBackoff=COLLISION\_AVOID

```

if(Whois & Burst)
    if(WhichBackoff == COLLISION_AVOID)
         $\beta_{\text{Source},(i)} = \text{rand}(0, \alpha)$ 
    else
         $\beta_{\text{Source},(i)} = \text{Adaptive\_Beta}$ 
    if( $\text{Thresh}_{CA} < \beta_{\text{Source},(i)}$ )
        Encapsulate  $\beta_{\text{Source},(i)}$  in packet header
        Adaptive Backoff( $\beta_{\text{Source},(i)}$ )
        Send_Packet()
    else
         $\beta_{\text{Source},(i)} = \text{Thresh}_{CA} + \beta_{\text{Source},(i)}$ 
        Encapsulate  $\beta_{\text{Source},(i)}$  in packet header
        Collision Avoidance Backoff( $\beta_{\text{Source},(i)}$ )
        Send_Packet()
else
    Previous Scheme
    
```

Send\_Packet()

Input:  $\beta_{\text{Source},(i)}$ , Output:  $T_{\text{Source},\text{timeout}(i)}$

```

if( $\alpha < \beta_{\text{Source},(i)} + \text{PACKET\_SIZE}$ )
     $T_{\text{Source},\text{timeout}(i)} = \beta_{\text{Source},(i)} + \text{PACKET\_SIZE}$ 
else
     $T_{\text{Source},\text{timeout}(i)} = \alpha$ 
    
```

Input:  $T_{\text{Source},\text{timeout}(i)}$  or  $\beta_{\text{Relay},(i)}$

Output: Signal to the Application Layer

$\text{Thresh}_{CD} = \text{Thresh}_{CA} / 2$

```

for( $T_{\text{Source},\text{timeout}(i)}$ )
    if((Not receiving overhearing packet) && (Timeout))
        Signal to the Application Layer
    else
        Stop timeout timer
        if( $\beta_{\text{Relay},(i)} == \text{Thresh}_{CD}$ )
            WhichBackoff = COLLISION_AVOID
        else
            WhichBackoff = ADAPTIVE
            Adaptive_Beta =  $\beta_{\text{Relay},(i)}$ 
        Signal to the Application Layer
    
```

그림 13 PIG 제어 알고리즘의 의사 코드

$\text{Thresh}_{CD}$ 를 비교하여 이전 노드가  $\text{Thresh}_{CD}$ 에 걸렸는지 판단한 후 *WhichBackoff*를 적절하게 설정한다.

그 이후 포워딩 과정을 거치게 된다. 포워딩은 목적지 주소를 확인하여 자신에게 온 패킷일 경우 이전 순서 번호(SN)와 현재 순서 번호를 비교하여 연속적인 패킷을 다음 노드로 순서적으로 전송할 수 있다. 그렇지 않은 경우 패킷을 폐기한다. 따라서 근원지 노드를 제외하고 패킷을 릴레이 하는 노드는 엿듣기 패킷을 MAC 레벨에서 즉시 차단하게 되고, 자신에게 오는 패킷만을 다음 노드로 포워딩하게 된다. 이러한 방식을 지원하기 위

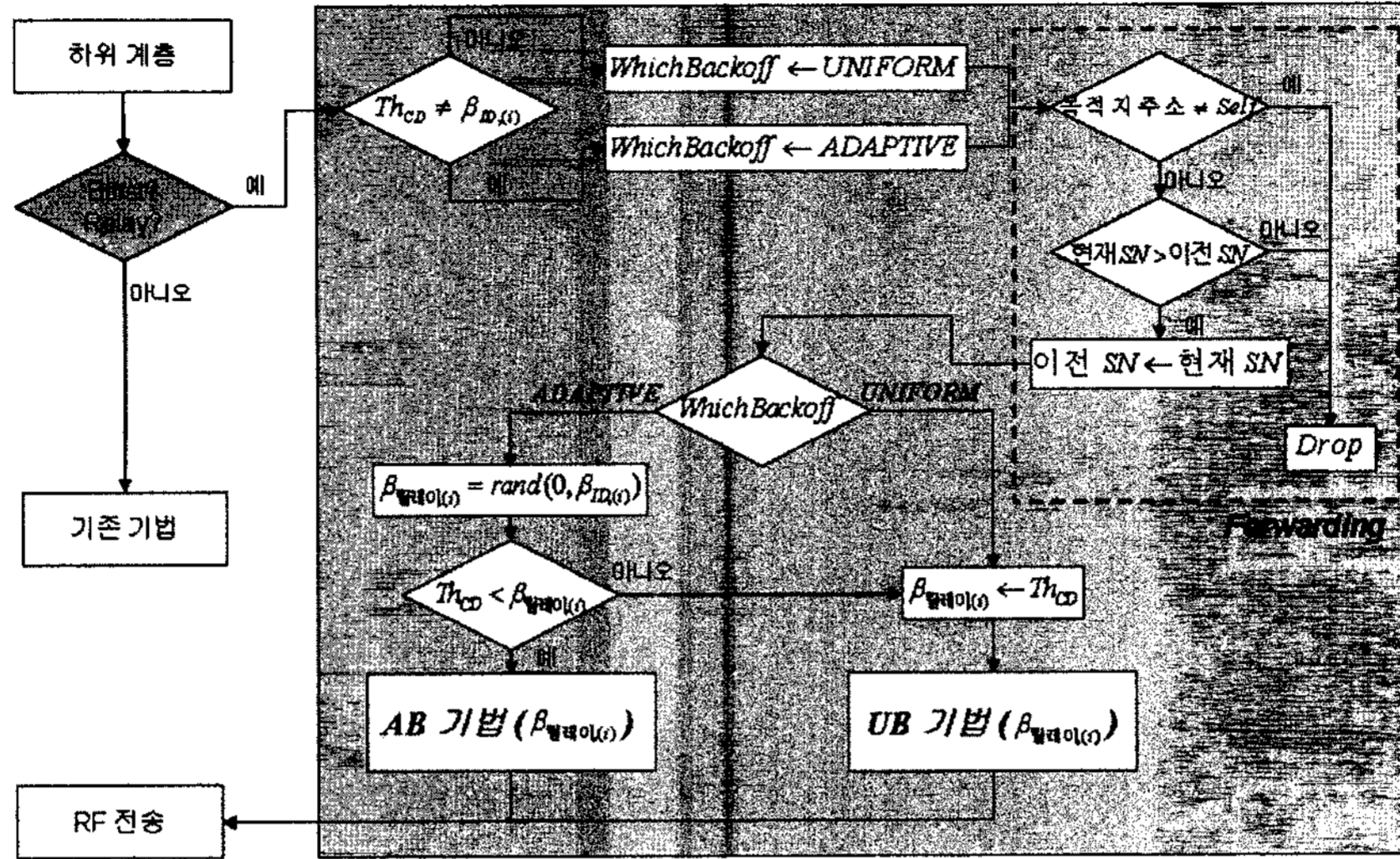


그림 14 MF 알고리즘

```

Input: Packet from the Lower Layer( $\beta_{ID(i)}$ )
Output: Determination of the WhichBackoff
Whois=Forwarder, WhichBackoff=ADAPTIVE
if(Whois & Burst)
    if(ThreshCD ==  $\beta_{ID(i)}$ )
        WhichBackoff = UNIFORM
    else
        WhichBackoff = ADAPTIVE
Forwarding()
else
    Previous Scheme

Forwarding( )
Input: Packet from the Lower Layer( $\beta_{ID(i)}$ ), WhichBackoff
Output: MAC-level Forwarding
ThreshCD = ThreshCA / 2
if(Rcvd ID == Self)
    if(Current SN > Previous SN)
        Previous SN = Current SN // Remember SN
    else
        Drop
if(WhichBackoff == ADAPTIVE)
     $\beta_{Relay(i)} = rand(0, \beta_{ID(i)})$  // Adaptive Backoff
    if(ThreshCD <  $\beta_{Relay(i)}$ )
        Encapsulate  $\beta_{Source(i)}$  in packet header
        Adaptive Backoff( $\beta_{Source(i)}$ )
        RF Transmission
    else
         $\beta_{Relay(i)} = Thresh_{CD}$ 
        Encapsulate  $\beta_{Source(i)}$  in packet header
        Uniform Backoff( $\beta_{Relay(i)}$ )
        RF Transmission
    
```

그림 15 MF 알고리즘의 의사 코드

해서는 근원지에서 최종 목적지로 가는 경로(routing route)가 결정되어 있어야 한다. 이를 위해 본 논문에서는 경로가 이미 결정되었다고 가정한다.

포워딩이 결정된 패킷은 설정된 WhichBackoff를 기반으로 AB와 UB로 전송된다. 만일 WhichBackoff가 UNIFORM일 경우 패킷 전송에 사용될  $\beta_{릴레이(i)}$ 를 Thresh<sub>CD</sub>로 갱신한 후, UB로 패킷을 전송하게 된다. 따라서 이러한 패킷을 받은 노드는 이전 노드가 Thresh<sub>CD</sub>에 걸렸음을 인지할 수 있게 된다. 반면에, WhichBackoff가 ADAPTIVE일 경우 최대 백오프 범위가 아닌  $\beta_{ID(i)}$ 를 기반으로  $\beta_{릴레이(i)}$ 가 랜덤하게 선택되고, 그 백오프 값을 Thresh<sub>CD</sub>와 비교하여 Thresh<sub>CD</sub>보다 큰 경우 기본적인 AB로 패킷을 전송하게 된다. 그렇지 않은 경우 UB와 동일한 방식으로 패킷을 전송하게 된다.

지금까지 설명한 MF 알고리즘을 의사 코드로 정리하였다. 이러한 의사 코드를 바탕으로 MICAZ의 MAC 계층에 제안하는 알고리즘을 구현하였다. MF 알고리즘은 수신된 패킷에 담긴 백오프 값을 기반으로 패킷을 MAC 레벨에서 신속하게 전송하는 알고리즘이다. 또한 Thresh<sub>CD</sub>를 통해 이웃 노드의 백오프를 인지할 수 있기 때문에 패킷 충돌 없이 패킷을 다음 노드로 포워딩할 수 있다.

#### 4. 실험 및 성능 평가

본 장에서는 새롭게 제안하는 PIGAB(Packet Interval Gap based on Adaptive Backoff) 프로토콜의 성능 평가를 위해 Matlab를 이용한 시뮬레이션과 TinyOS[14] 기반의 MICAZ에 PIGAB을 구현하여 실제 실험을 수

행하였다.

4.1 실험 환경

새로 제안한 PIGAB 프로토콜의 성능을 평가하기 위해 그림 16과 같은 무선 센서 네트워크를 구성하여 홉을 늘려가며 성능 실험을 수행하였다.

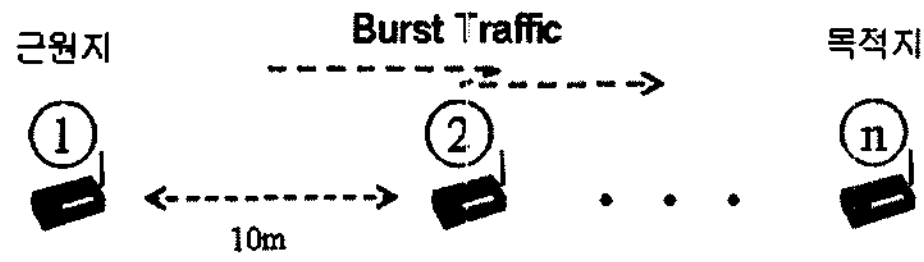


그림 16 실험 망 구성

표 1 실험 환경

센서 플랫폼 (sensor platform)	• TinyOS 기반의 MICAZ : CC2420 RF 모듈
응용 계층	• 데이터 크기(110Bytes) =헤더(7Bytes)+페이로드(103Bytes) • 버퍼링 기법 고려하지 않음
라우팅 계층	• 고정 라우팅 사용
MAC 계층 (CC2420 Radio)	• RF 전송 파워 : -10dBm으로 고정 사용 • ACK 사용하지 않음

PIGAB 프로토콜의 성능 평가를 위해 표 1의 실험 환경을 기반으로 2.2절에 기술한 기존 방식과 본 논문에서 제안하는 방식을 비교하였다. 실험에 사용된 센서 플랫폼으로는 TinyOS 기반의 MICAZ을 사용하였다. 이 플랫폼에서 사용하는 CC2420[15] RF 모듈은 2.4Ghz 주파수 대역을 사용하며, 250Kbps의 전송률을 가지고 있다. 그리고 한 패킷의 데이터 크기는 110B로서 데이터 헤더(header)는 7B, 페이로드(payload)는 103B로 구성하였다. 따라서 응용 계층에서는 이러한 데이터를 연속적으로 버스트하게 생성하여 이미 결정된 경로(routing route)를 통해 전송하게 된다. 또한 RF 전송 파워는 기본적으로 -20~0dBm의 범위를 가지고 있으며, 실험에서는 거리상의 제약으로 인해 패킷 전송을 위한 파워를 -10dBm으로 고정하여 사용하였다. 그리고 패킷 전송의 신뢰성을 위해 사용될 수 있는 MAC 레벨의 지역적 ACK 기법(local ACK)은 사용하지 않았다. 이러한 실험 환경을 기반으로 제안된 PIGAB 프로토콜의 성능 실험을 수행하였다.

4.2 성능 평가를 위한 파라미터 설정

본 절에서는 기존 방식의 응용 계층에서 사용될 패킷 전송 시기를 결정하기 위한 실험을 수행하였다. 그리고 제안된 백오프 기법과 기존 방식의 백오프 기법을 비교하기 위해 패킷 전송에 사용된 MAC 계층에서의 백오프 변화를 시뮬레이션을 통해 확인하였다.

4.2.1 기존 방식의 응용 계층에서의 패킷 전송 시기 (interval)

제안된 PIGAB 프로토콜과의 효과적인 비교를 위해 기존 방식의 응용 계층에서 사용될 최적의 패킷 전송 시기를 결정하는 실험을 수행하였다. 본 실험은 단일 홉 환경에서 전송 시기를 변화시키면서 1000개의 패킷에 대한 수신 성공률과 전송 소요 시간을 측정하였다.

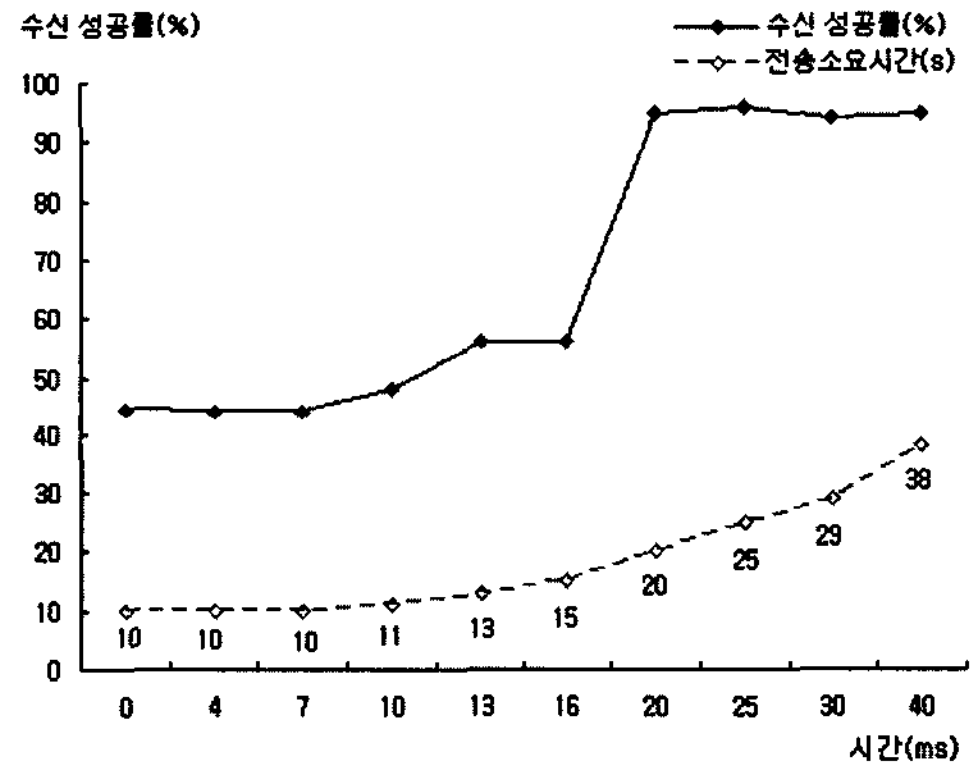


그림 17 최적의 패킷 전송 시기

그림 17은 그 결과를 나타낸 것이다. 최적의 패킷 전송 시기는 높은 수신 성공률과 낮은 전송 소요 시간을 동시에 만족하는 곳에서 결정된다. 따라서 기존 방식의 응용 계층에서 사용될 최적의 패킷 전송 시기는 20ms로 결정되며, 이러한 시기로 패킷을 전송할 경우 높은 수신률과 낮은 전송 시간을 갖게 된다. 본 실험 결과를 통해 기존 방식의 패킷 전송 시기를 20ms로 하여 PIGAB 프로토콜과의 모든 성능 비교 실험을 수행하였다.

4.2.2 MAC 계층에서의 백오프 변화

경쟁 기반 MAC 프로토콜은 랜덤 백오프와 물리 계층에서의 CCA(Clear Channel Assessment)를 기반으로 상위에서 받은 패킷을 실제 전송하게 된다. MAC 계층에서의 랜덤 백오프 기법은 선택적으로 구현될 수 있다. 하지만 경쟁을 최소화하기 위해 랜덤 백오프 후 패킷을 전송하는 것이 일반적이다. 그러나 무선 센서 네트워크 환경에서 연속적인 패킷의 버스트 데이터를 멀티홉으로 전송하기에 기존의 백오프 기법은 바람직하지 않다. 그 첫 번째 이유는 2.3절에서 원인 분석을 통해 이미 확인한 바 있다.

본 절에서는 Matlab으로 수행된 시뮬레이션을 통해 두 번째 이유를 제시하고, 기존 기법보다 제안된 백오프 기법이 백오프 변화에 있어 버스트 데이터를 전송하는데 적합함을 확인한다. 이를 위해 3홉 환경에서 50개의 패킷 전송에 따른 백오프 값의 변화를 각각 확인하였다.

그림 18은 기존 기법에 대한 백오프 값의 변화를 나타낸 것이다. 각 패킷들이 최대 백오프 범위에서 랜덤하게 선택되는 것을 확인할 수 있다. 이것은 2.3.1절에서 실험 결과로 제시한 홉에 따른 패킷 손실의 원인이 된다. 반면에 그림 19는 제안된 기법에 대한 백오프 값의 변화를 나타낸 것이다. 그림 18과 비교하여 백오프 값이 가공되고 정제되었음을 확인할 수 있다.

기존 기법과 제안된 백오프 기법은 기본적으로 랜덤 성향을 갖는다. 하지만 기존 백오프 기법은 연속적인 버스트 데이터를 전혀 고려하지 않았다. 하지만 제안된 백오프 기법은  $Thresh_{CD}$ 와  $Thresh_{CA}$ 를 기반으로 상황에 따라 동적으로 백오프 값을 조절한다. 이러한 이유 때문에 멀티홉 환경에서 버스트 데이터를 전송하는데 적합함을 본 시뮬레이션을 통해 확인할 수 있었다.

PIGAB 프로토콜은 제안된 백오프 기법을 상황에 따라 동적으로 적용하기 위해  $Thresh_{CD}$ 와  $Thresh_{CA}$ 를 정의하고 있다. 이러한 임계값은 식 (5)의 조건하에서  $Thresh_{CA}$ 은 식 (6)으로  $Thresh_{CD}$ 은 식 (7)으로 설정하여 PIGAB 프로토콜의 성능 평가를 위해 모든 시뮬레이션과 실험에 사용되었다. 그림 19에서 이렇게 설정된 임계값을 기반으로 백오프 값을 조절하는 것을 확인할 수 있다.

$$0 < Thresh_{CD} < Thresh_{CA} < \alpha \tag{5}$$

$$Thresh_{CA} = \frac{\alpha}{2} \tag{6}$$

$$Thresh_{CD} = \frac{Thresh_{CA}}{2} \tag{7}$$

### 4.3 PIGAB 프로토콜 성능 실험

본 실험은 PIGAB 프로토콜의 성능을 평가하기 위한 것이다. 멀티홉 환경에서 버스트 데이터의 전송률(throughput)과 그에 따른 수신 성공률에 대한 성능을

확인한다. 그리고 멀티홉에서의 패킷 전송 지연(latency)에 대한 성능을 확인한다.

#### 4.3.1 단일 홉(Single hop)에서의 대역폭 측정

우선, 250Kbps를 갖는 CC2420에 대한 실제 가용 대역폭을 확인하기 위해 단일 홉 환경에서 1000개의 패킷을 주입하여 그에 따른 전송률을 측정하였다. 그림 20의 가용 대역폭은 95% 이상의 수신률일 때의 결과를 나타낸 것이며, 제안된 PIGAB 프로토콜의 전송률이 향상됨을 확인할 수 있다. 그 이유는 엿듣기 패킷의 백오프 값과 타임아웃을 기반으로 동적으로 전송률을 조절하기 때문이다.

#### 4.3.2 멀티홉에서의 수신 성공률 성능 평가

본 실험에서는 멀티홉 환경에서 각 노드에서의 패킷 수신 성공률에 대한 성능을 평가하였다. 4.1절에서 기술한 실험 환경을 기반으로 100개의 패킷을 버스트하게 네트워크로 주입하여 기존 방식과 제안된 PIGAB 프로토콜을 비교하였다.

그림 21은 비교 결과를 나타낸 것이다. 기존 방식보다 제안된 프로토콜의 수신률이 홉이 증가할수록 상당히 향상되었음을 확인할 수 있다. 그 이유는 홉이 증가함에 따라 기존 방식은 감춰진 노드에 대한 고려가 없기 때문에 충돌로 인해 수신률이 급격히 저하되지만, 제안된 PIGAB 프로토콜은 이러한 문제를 고려하여 설계되었기 때문에 좋은 성능을 보인다. 본 실험의 결과를 통해 홉이 증가 하더라도, 제안된 프로토콜은 근원지 노드에서 감춰진 노드의 전송 상황을 인지하여 다음 패킷을 전송하기 때문에 충돌을 줄여 수신률의 향상을 가져올 수 있었다.

본 실험의 결과를 통해 멀티홉 환경에서 기존 방식을 사용할 경우 전송 가능한 홉 수를 확인할 수 있었다. 3홉 이상에서 버스트 데이터에 대한 수신률이 급격히 저하되며, 4홉에서 50% 미만의 수신률을 보이고, 5홉에서

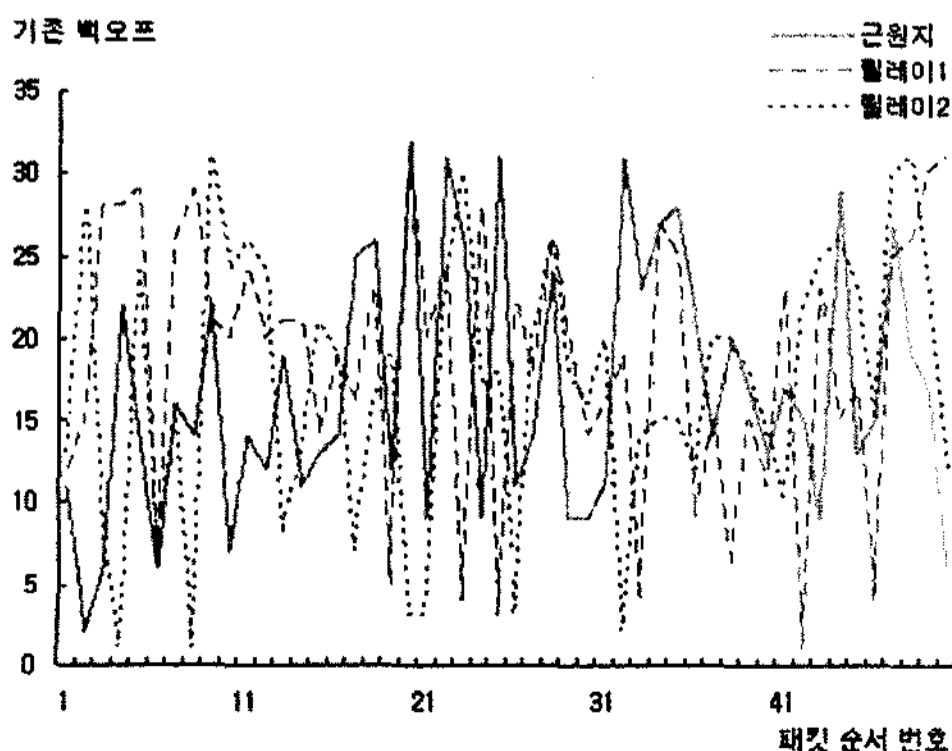


그림 18 기존 백오프 기법

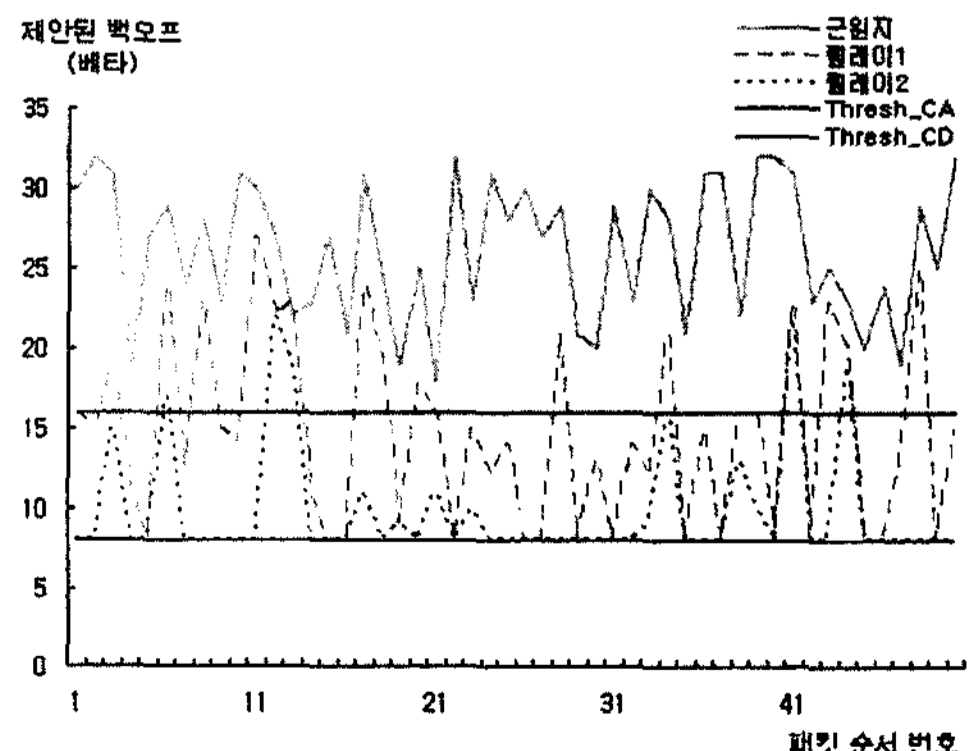


그림 19 제안된 백오프 기법

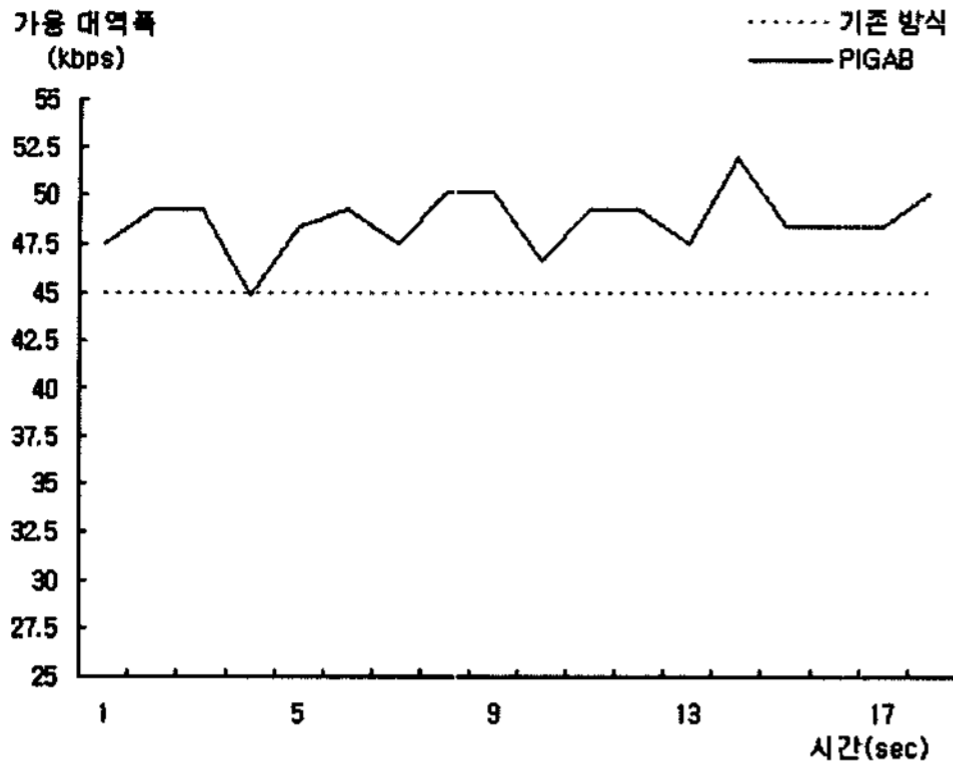


그림 20 단일 홉에서의 가용 대역폭

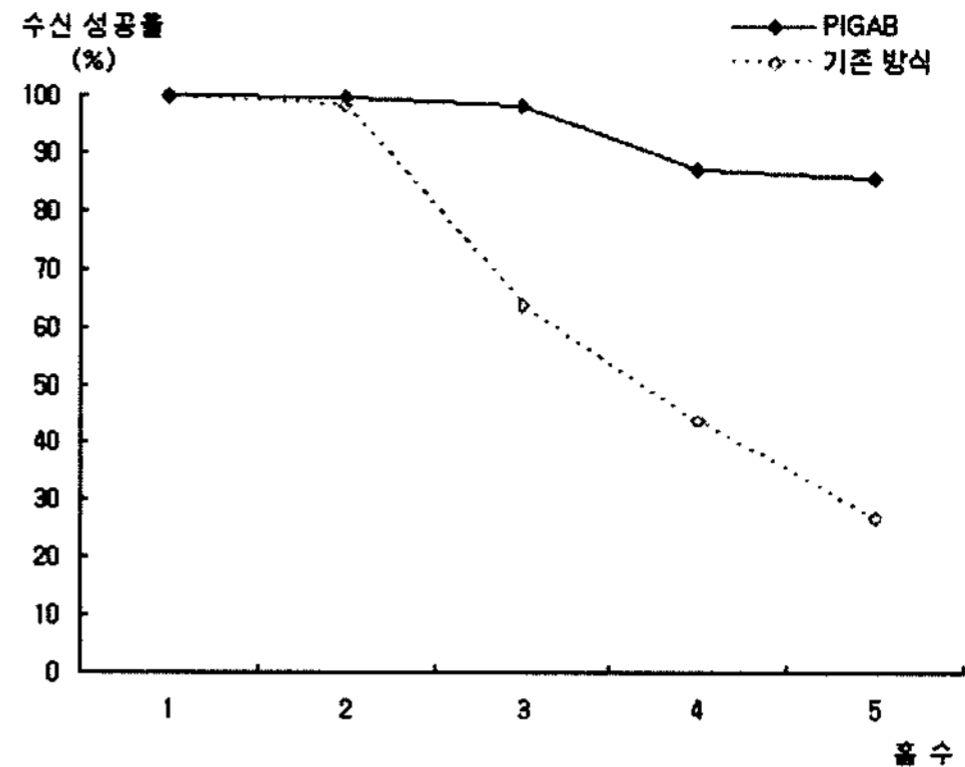


그림 21 멀티홉에서의 수신 성공률 비교

20~30%의 상당히 낮은 수신률을 보였다. 따라서 5홉 이상으로 실험하는 것은 무의미하다고 판단하여, 4.3.3과 4.3.4절에서는 4홉까지의 실험 결과를 바탕으로 성능 평가를 수행하였다.

4.3.3 전송률(Throughput) 성능 평가

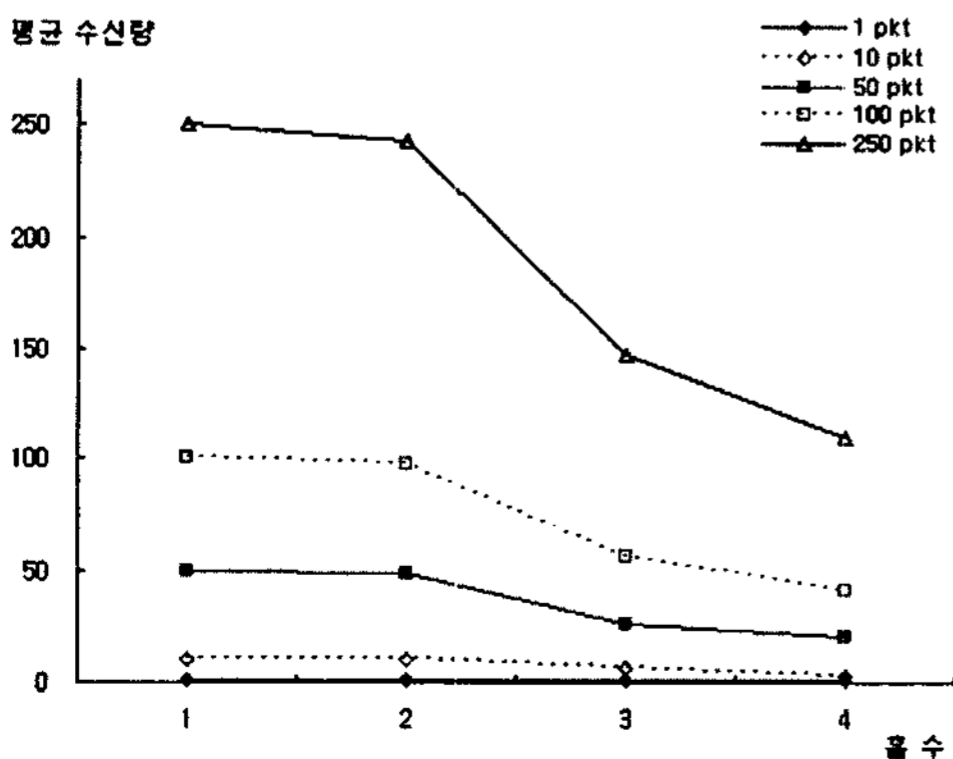
본 실험에서는 멀티홉 환경에서 전송 패킷에 대한 평균 수신량(throughput)과 각 노드에서의 패킷 수신 성공률에 대한 성능을 평가하였다. 4.1절에서 기술한 실험 환경을 기반으로 패킷 주입량을 변화시켜가며 기존 방식과 제안된 PIGAB 프로토콜을 비교하였다.

그림 22는 패킷 주입량에 따라 홉에서의 평균 수신량을 나타낸 것이다. 그림 22의 (a)는 기존 방식에 대한 결과이며, 2홉까지 좋은 전송률을 보이지만 감춰진 노드가 생기는 3홉부터 전송률이 전체적으로 감소하는 것을 확인할 수 있다. 특히, 전송 패킷량이 증가함에 따라 급격히 감소되었다. 이를 통해 연속적인 버스트 패킷의 증가는 전송률을 감소시킴을 알 수 있다. 기존 방식에서의 이러한 성능 저하의 원인은 멀티홉 환경에서 버스트 데

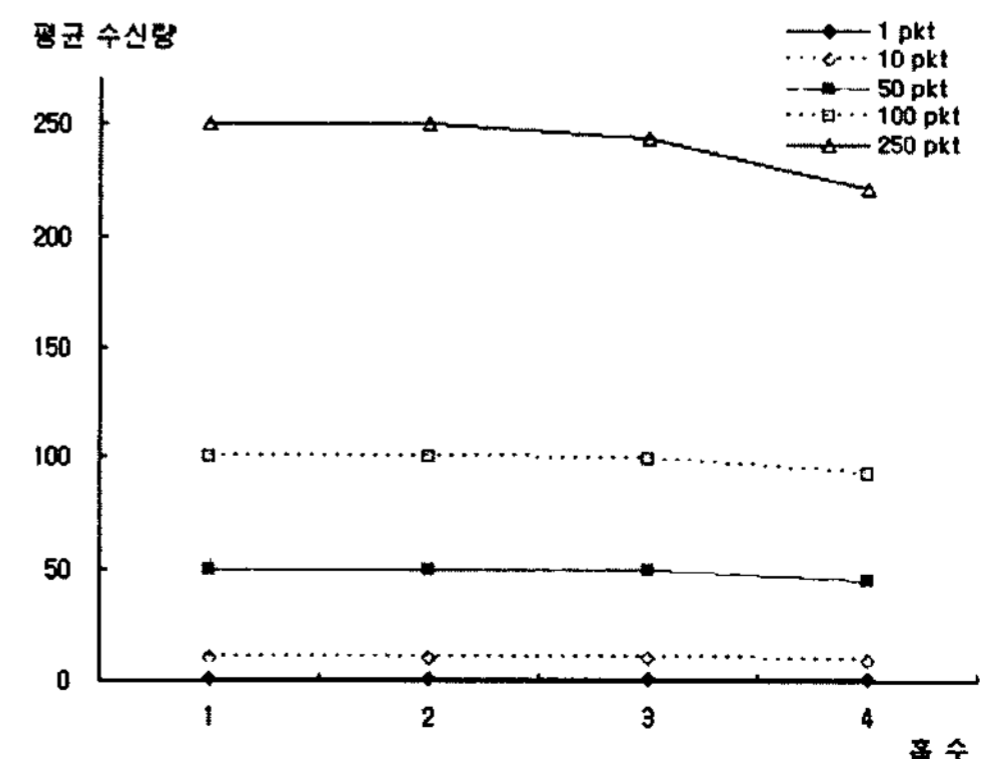
이타에 대한 고려를 하지 않았기 때문이다.

반면에, 그림 22의 (b)는 제안된 PIGAB 프로토콜에 대한 결과이며, 기존 방식과 비교하여 전체적으로 안정적이고 높은 전송율을 보여준다. 특히 홉이 증가하더라도 증가하는 전송 패킷량에 대해 각 노드에서 안정적으로 패킷을수신하는 것을 확인할 수 있다. 이러한 성능 향상은 멀티홉 환경에서 버스트 데이터를 고려하여 PIGAB 프로토콜이 설계되었기 때문이다. 근원지 노드에서는 엿듣기 패킷에 담긴 백오프 값과 타임아웃을 기반으로 다음 패킷의 전송 시기를 결정하고, 제안하는 AB와 CAB 기법으로 패킷을 전송하게 된다. 그리고 수신 노드에서는 수신 패킷에 담긴 백오프 값을 기반으로 제안하는 AB와 UB 기법으로 MAC 레벨에서 패킷을 다음 노드로 포워딩하게 된다. 이러한 방식으로 인해 멀티홉 환경에서 버스트 데이터를 안정적으로 전송할 수 있다.

표 2는 각 노드에서의 패킷 수신 성공률에 대한 데이터를 나타낸 것이다. 기존 방식과 비교하여 제안된 PIGAB 프로토콜이 상당히 높은 수신률을 보였다. 그림



(a) 경쟁 기반 프로토콜

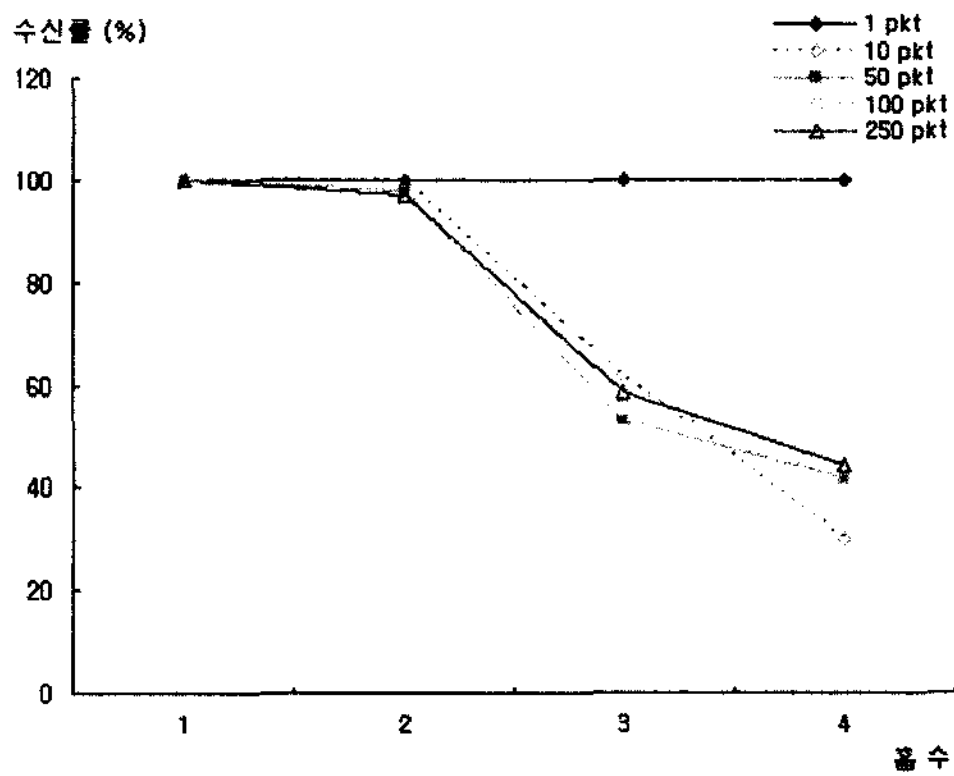


(b) PIGAB 프로토콜

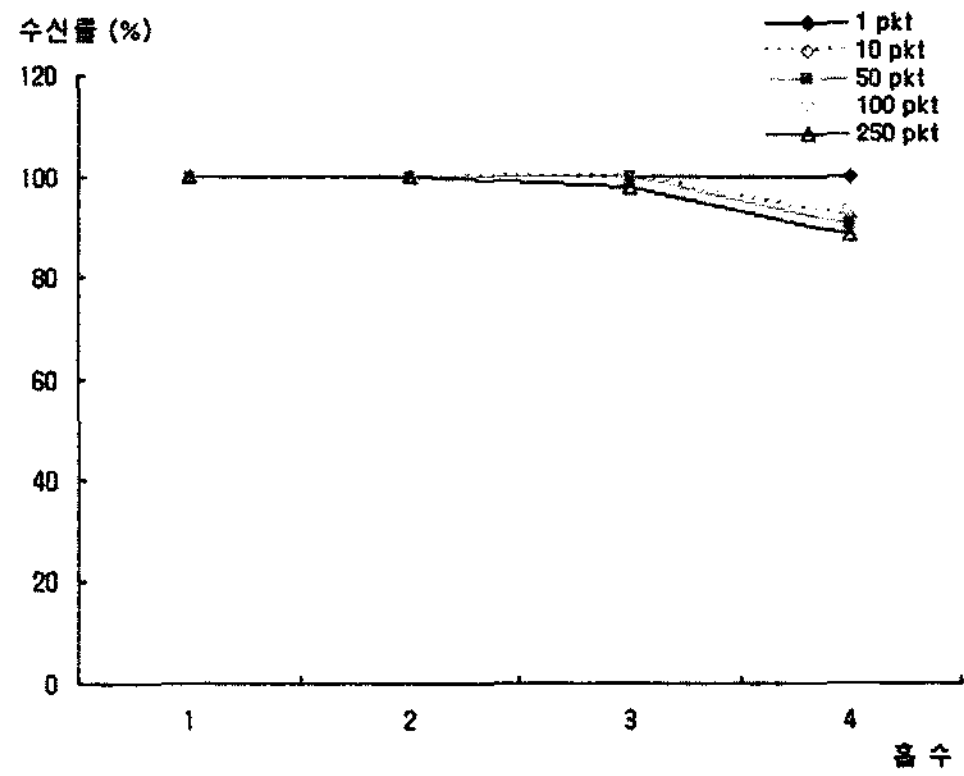
그림 22 평균 전송률 비교

표 2 패킷 수신 성공률 (단위: %)

패킷 \ 홉수	1		2		3		4	
	기존방식	PIGAB	기존방식	PIGAB	기존방식	PIGAB	기존방식	PIGAB
1	100	100	100	100	100	100	100	100
10	100	100	100	100	62	100	30	92
50	100	100	98	100	52.8	100	41.6	90.8
100	100	100	97.6	100	55.8	99	41.2	93
250	100	100	97.2	100	58.8	97.6	44	88.4



(a) 경쟁 기반 프로토콜



(b) PIGAB 프로토콜

그림 23 수신률 비교

23은 이러한 결과를 그래프로 나타낸 것이다.

### 4.3.3 지연(Latency) 성능 평가

본 실험에서는 멀티홉 환경에서 전송 패킷의 평균 지연 시간에 대한 성능을 평가하였다. 4.1절에서 기술한 실험 환경을 기반으로 50개의 패킷을 주입하여 홉에 따라 패킷 당 평균 지연 시간을 측정하였다. 그림 24는 그 결과를 나타낸 것이다. 기존 방식에 비해 PIGAB 프로토콜의 성능이 향상되었다. 그 이유는 상황에 따라 동적으로 작용하는 백오프 기법과 수신단에서의 MF 알고리즘에 의한 것이다. 표 3은 지연 시간에 결과를 자세히 나타낸 것이다.

표 3 패킷 당 평균 지연 시간 (단위: ms)

방식 \ 홉수	1	2	3	4
기존 방식	9.86	24.56	42.70	57.05
PIGAB	7.12	21.99	31.95	42.75

## 4. 결론

최근 들어 다양한 응용에 적용 가능한 무선 센서 네트워크 기술에 대한 연구가 활발히 진행되고 있다. 아직은 많은 부분들이 표준화되지 않은 영역으로 관련 연구의 확산 측면에서 다소 어려움이 있지만, 새로운 응용을 창조하고 그 응용을 지원하기 위한 프로토콜을 설계하는 측면에서 도전적인 분야라고 할 수 있다.

지금까지의 센서 네트워크 응용은 단일 형태의 비교적 단순한 센싱 데이터만을 취급해 왔고, 이를 지원할 수 있도록 통신 프로토콜들이 설계되어 왔다. 또한 적은 양의 파워로 장시간 동작하는 특징 때문에 프로토콜은 에너지 효율성에 편중되어 설계되어 왔다. 하지만 최근 카메라(camera)나 이미지(image) 센서가 장착된 새로운 형태의 플랫폼(platform)들이 연구 개발됨에 따라 데이터 크기가 큰 새로운 유형의 트래픽(traffic)이 생겨나게 되었다. 이러한 트래픽은 일시적으로 발생 가능하며 여러 패킷으로 이루어진다. 따라서 종전의 데이터 형태뿐만 아니라 이러한 버스트(burst) 데이터를 멀티홉 환경

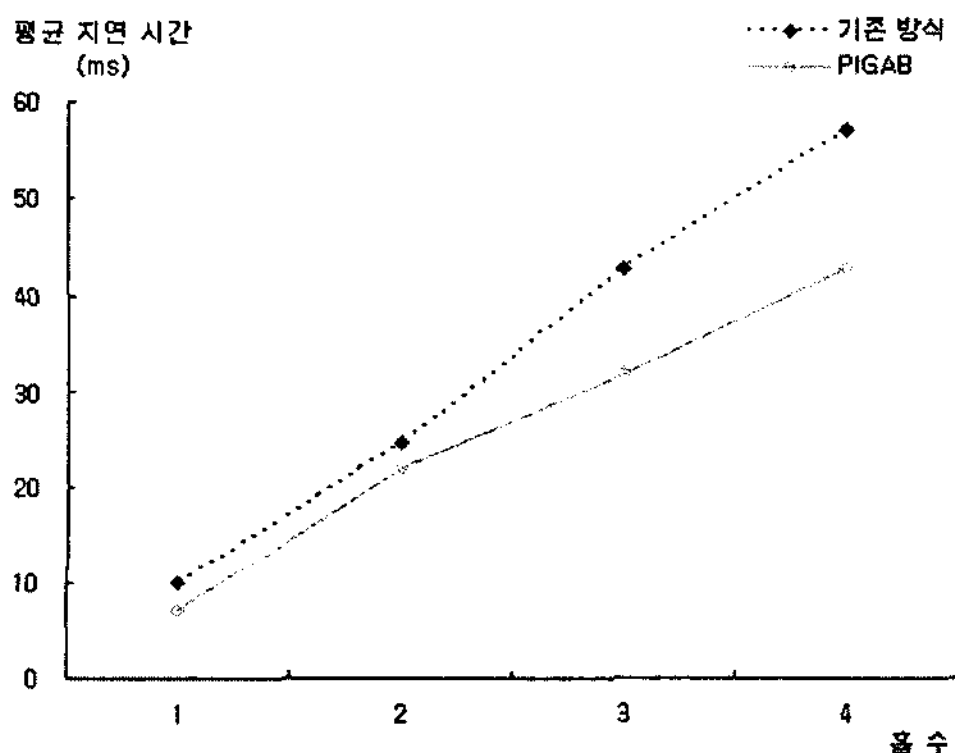


그림 24 평균 지연 시간 비교

으로 전송할 수 있는 프로토콜이 필요하다.

하지만 이를 위한 연구가 지금까지 거의 진행되지 않았고, 일부 연구에서 문제점을 거론하고 있지만 구체적인 알고리즘을 제시하지 못하고 있다. 기본적으로 멀티홉 환경에서 버스트 데이터를 전송하기 위해서는 감춰진 노드로 인한 패킷 손실 문제를 해결해야 한다. 이를 해결하기 위한 기존의 RTS/CTS 기법은 프로토콜 오버헤드(overhead)가 크기 때문에 한계를 갖는다.

본 논문에서는 이러한 문제를 해결하여 버스트 데이터를 멀티홉으로 안정적이고 신속하게 전송하는 PIGAB(Packet Interval Gap based on Adaptive Backoff) 프로토콜을 제안하였다. 경쟁 기반 프로토콜인 PIGAB은 근원지 노드에서 동작하는 PIG(Packet Interval Gap) 제어 알고리즘과 릴레이 노드에서 동작하는 MF(MAC-level Forwarding) 알고리즘으로 구성되며, 새롭게 제안된 AB(Adaptive Backoff), CAB(Collision Avoidance Backoff), 그리고 UB(Uniform Backoff)를 기반으로 동작한다. 제안된 PIGAB 프로토콜은 이러한 알고리즘과 기법을 통해 감춰진 노드의 전송 시기를 인지하여 패킷 마다의 전송 시기를 조절함으로써 멀티홉 환경에서 겪는 기본적인 문제를 해결할 수 있다. 시뮬레이션과 실제 실험을 통해 PIGAB 프로토콜이 기존 방식에 비해 멀티홉 환경에서 버스트 데이터를 안정적이고 신속하게 전송하는 것을 확인 할 수 있었다.

향후 연구 과제로는 여러 근원지에서 발생하는 버스트 트래픽들이 경쟁하는 상황에서 제안된 PIGAB 프로토콜에 대한 성능 검증이 필요하며, 이러한 상황에서 여러 버스트 트래픽들을 공정하게 전송할 수 있도록 하는 방법에 대한 연구가 수행되어야 할 것이다.

## 참 고 문 헌

- [1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "A Survey on Sensor Networks," IEEE Communications Magazine, August 2002.
- [2] Akyildiz, I.F., Melodia, T., and Chowdry, K.R., "A Survey on Wireless Multimedia Sensor Networks," Computer Networks Elsevier Journal, March 2007.
- [3] Jingbin Zhang, Gang Zhou, Sang H. Son, John A. Stankovic, "Ears on the Ground: An Acoustic Streaming Service in Wireless Sensor Networks," The 5th International Conference on Information Processing in Sensor Networks(IPSNS 2006), April 2006.
- [4] Wei Ye, John Heidemann and Deborah Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," The 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2002), June 2002.
- [5] Alec Woo and David Culler, "A Transmission Control Schemes for Media Access in Sensor Networks," The 7th Annual International Conference on Mobile Computing and Networking(Mobicom 2001), July 2001.
- [6] Phil Karn, "MACA - A New Channel Access Method for Packet Radio," The 9th ARRL Computer Networking Conference, September 1990.
- [7] Joe Polastre and David Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," The 2nd ACM Conference on Embedded Networked Sensor Systems(SenSys 2004), November 2004.
- [8] M. Buettner, G. Yee, E. Anderson, R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," The 4th ACM Conference on Embedded Networked Sensor Systems(SenSys 2006), October 2006.
- [9] L.F.W. van Hoesel, P.J.M. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches," The 1st International Workshop on Networked Sensing Systems(INSS 2004), June 2004.
- [10] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," The 1st ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), November 2003.
- [11] K. Langendoen and G. Halkes, "Energy-Efficient Medium Access Control," Book Chapter in the Embedded Systems Handbook, R. Zurawski (editor), CRC Press, August 2005.
- [12] Crossbow MICAZ motes, <http://www.xbow.com>
- [13] The MathWorks Inc., Matlab version 7.0.1, <http://www.mathworks.com>
- [14] TinyOS, <http://www.tinyos.net>
- [15] Chipcon CC2420 radios, <http://www.chipcon.com>



노 태 호

2005년 광운대학교 전자공학부 학사. 2007년 광운대학교 전자통신공학과 석사. 2007년~현재 이노와이어리스 정보통신연구소 T&M S/W개발팀. 관심분야는 센서 네트워크, 임베디드시스템, ad-hoc 네트워크

정 광 수

정보과학회논문지 : 정보통신  
제 35 권 제 2 호 참조