

논문 2010-47TC-6-2

LR-WPAN에서 충돌회피를 위한 동적 채널할당 알고리즘

(The Dynamic Channel Allocation Algorithm for Collision Avoidance in LR-WPAN)

임 정 섭*, 윤 완 오*, 서 장 원*, 최 한 림*, 최 상 방**

(Jeong-Seob Lim, Wan-Oh Yoon, Jang-Won Seo, Han-Lim Choi, and Sang-Bang Choi)

요 약

많은 노드와 넓은 지역을 커버하기 위해 클러스터-트리 형태로 구성된 모니터링 네트워크는 싱크 주변에 트래픽이 집중된다. IEEE 802.15.4가 이러한 센서 네트워크에 적용 되었을 때 싱크 주변지역에서의 트래픽 집중으로 인하여 데이터의 전송 지연이 길어지고, 데이터를 잃어버릴 가능성이 높다. 본 논문에서는 LR-WPAN에서 채널을 동적으로 할당하여 채널 사용률을 높이고, 데이터 전송 성공률을 높이는 알고리즘을 제안한다. 본 논문에서는 제안한 알고리즘의 성능평가를 위하여 IEEE 802.15.4의 액티브 구간길이를 다양하게 설정하여 실험하였다. 실험 환경은 클러스터-트리 형태의 모니터링 네트워크를 가정하고, 트래픽이 집중되는 지역인 싱크와 1홉 거리에서 발생하는 트래픽을 분석했다. 실험결과 트래픽이 많을수록 DCA가 우수한 성능을 보였다. 트래픽이 가장 적은 상황에 대한 실험에서 IEEE 802.15.4는 최소한의 액티브길이인 14.40ms로 동작하여 90.3%의 데이터 전송 성공률을 보이고, DCA는 11.8ms의 액티브 구간을 유지하며 98.9%의 데이터 전송 성공률을 보였다.

Abstract

In the cluster-tree network which covers wide area network and has many nodes for monitoring purpose traffic is concentrated around the sink. There are long transmit delay and high data loss due to the intensive traffic when IEEE 802.15.4 is adapted to the cluster-tree network. In this paper we propose Dynamic Channel Allocation algorithm which dynamically allocates channels to increase the channel usage and the transmission success rate. To evaluate the performance of DCA, we assumed the monitoring network that consists of a cluster-tree in which sensing data is transmitted to the sink. Analysis uses the traffic data which is generated around the sink. As a result, DCA is superior when much traffic is generated. During the experiment assuming the least amount of traffic, IEEE 802.15.4, has the minimum length of active period and 90% data transmission success rate. However DCA maintains 11.8ms of active period length and results in 98.9% data transmission success rate.

Keywords : LR-WPAN, IEEE 802.15.4, Zigbee, Beacon-enabled, Sensor network

I. 서 론

일상생활이나 산업, 가정 등에서 발생하는 여러 가지 이벤트들을 감지하는 센서들은 지금까지 주로 주변의 상황을 센싱하고, 사용자가 센서로 가서 데이터를 확인하거나, 유선을 통해 단순한 데이터를 전송하는 방식으로 이용되었다. 그러나 센서들이 무선 네트워크로 연결

되면 설치비용의 절감과, 새로운 어플리케이션의 적용이 가능하다. 이를 위해서 IEEE에서는 무선 센서 네트워크를 위한 LR-WPAN(Low Rate-Wireless Personal Area Network)인 IEEE 802.15.4 표준을 제정했다. 무선 센서 네트워크는 고속의 데이터 전송보다는 낮은 비용으로 오랜 시간동안 동작할 수 있는 기술이 필요하기 때문에 IEEE 802.11 계열이나 IEEE 802.15.1 Bluetooth와는 다르게 저전력, 저비용, 저속의 특징을 갖는다.

LR-WPAN는 홈, 빌딩, 가전, 헬스케어 등의 응용에 쉽게 적용될 수 있고, 신뢰성 있는 데이터 전송을 위해서 다양한 네트워크 토폴로지(topology)를 지원한다. 스

* 학생회원, ** 평생회원, 인하대학교 전자공학과

(Dept. of Electronic Engineering Inha University)

※ 이 논문은 인하대학교의 지원에 의하여 연구되었음.
접수일자 : 2010년1월14일, 수정완료일 : 2010년5월31일

타 토폴로지는 근거리의 간단한 네트워크를 구성하기 위해서 사용되고, 피어 투 피어(peer-to-peer) 토폴로지는 인프라구조 없이 모든 노드가 통신 영역 내에 있는 노드와 연결이 가능한 네트워크를 구성한다. 클러스터-트리는 라우터 노드가 존재하는 멀티 홉으로 구성되어 보다 광범위한 네트워크를 구성할 수 있다. 이러한 클러스터-트리 토폴로지를 기반으로 토양의 상태나, 빌딩 자동화, 도로상황, 여러 지역에서 동작중인 기계의 상태 모니터링 등의 무선 센서 네트워크를 구성할 수 있다 [1~4].

클러스터-트리 구조의 모니터링 네트워크는 넓은 지역을 커버하고, 많은 노드들이 참여한다. 이러한 네트워크에서 센싱 데이터는 싱크로 전달되기 때문에 싱크 주변지역으로 트래픽이 집중된다. IEEE 802.15.4는 트래픽이 적은 상황을 고려하여 설계되었기 때문에 이와 같은 네트워크의 싱크 주변에서 긴 전송지연을 발생시키고, 패킷 전송이 포기되거나 충돌로 인한 데이터 손실이 많아진다^[5~8].

무선 상황에서 충돌을 회피하는 방법은 RTS/CTS(Request To Send/Clear To Send)를 사용하는 폴링방식이 있다. RTS/CTS방식은 데이터를 전송하기 전에 RTS프레임을 보내고, 수신자로부터 CTS가 브로드캐스팅 되어 채널을 예약하는 방식이다. 하지만 이 방법은 RTS와 CTS를 교환하는데 오버헤드가 크기 때문에 LR-WPAN에 사용하는데 적절하지 않다^[9].

무선 충돌을 회피하는 또 다른 방법은 CSMA-CF(Carrier Sensing Multiple Access-Collision Free)가 있다. CSMA-CF는 히든 터미널 때문에 발생하는 충돌을 줄이는 알고리즘으로 코디네이터(coordinator)는 히든 터미널들을 서로 다른 그룹으로 묶어서 그룹마다 서로 다른 범위의 슬롯을 할당한다. CSMA-CF는 코디네이터에 참여한 자식노드가 많은 경우 효율적이고, 히든노드들 사이에 위치하는 노드들이 히든 터미널 충돌을 감지하여 코디네이터에 보고해야하는 오버헤드가 발생한다^[10].

본 논문에서는 데이터전송에 GTS(Guaranteed Time Slot)를 이용하고 트래픽을 예측하여 동적으로 채널을 할당하는 알고리즘을 제안한다. 제안된 알고리즘은 네트워크의 트래픽을 CFP(Contention Free Period)구간으로 분산시키기 때문에 패킷의 충돌확률이 낮아지고, 전송지연이 짧아진다. 또한 네트워크의 트래픽을 예측하여 동적으로 액티브 구간 길이를 설정하므로 불필요한

전력소모가 작다.

본 논문의 구성은 다음과 같다. II장에서는 IEEE 802.15.4표준에 대해 살펴보고 III장에서는 본 논문에서 제안하는 동적 채널할당 알고리즘에 대해서 설명한다. IV에서는 제안된 알고리즘의 성능평가를 위한 실험 환경과 실험 결과를 IEEE 802.15.4표준과 비교분석하고 V장에서는 결론을 기술하였다.

II. IEEE 802.15.4 표준

무선 네트워크에서 IEEE 802.15.4는 비콘(beacon)과 슈퍼프레임구조(superframe structure)의 사용으로 네트워크를 동기화시켜 네트워크에 참여한 노드들의 에너지 소모를 줄일 수 있다. 슈퍼프레임은 비콘을 수신하면서 부터 시작되며, 비콘에는 다음 비콘이 전송되기까지의 시간인 BI(Beacon Interval)와 액티브 구간 길이인 SD(Slot Duration)가 포함되어 있다. 그림 1은 IEEE 802.15.4에서 사용하는 슈퍼프레임구조를 나타낸다.

슈퍼프레임은 크게 액티브 구간과 인액티브 구간으로 나뉜다. 액티브 구간에서 데이터전송이 이루어지고 인액티브 구간에서 노드들은 저전력 모드로 동작하여 에너지를 절약한다. 액티브 구간은 CAP(Contention Access Period)와 CFP로 구성된다. CAP구간은 경쟁구간으로써 데이터를 전송하고자 하는 노드는 전송을 시작하기 전에 CSMA-CA(Carrier Sensing Multiple Access-Collision Avoidance)를 수행해야하고, CFP구간은 전송 지연에 민감한 어플리케이션을 위한 폴링구간으로써 자신에게 할당된 슬롯에서 데이터를 전송한다.

slotted CSMA-CA는 IEEE 802.15.4에서 충돌을 회피하기 위한 기법으로 채널을 사용하기 전에 랜덤 백오프(random backoff) 슬롯만큼 기다리고, 두 번의 CCA(Clear Channel Assessment)가 성공해야 채널을 사용한다. CCA가 실패하면 노드는 현재 백오프 슬롯을 두 배로 늘리고 새로운 백오프 슬롯만큼 기다린 후 다시 CCA를 시도한다. IEEE 802.15.4는 기본으로 네

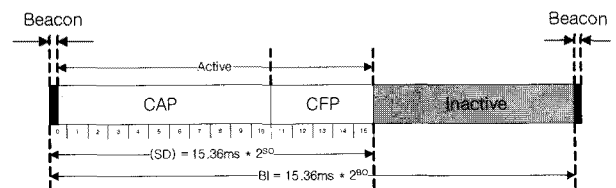


그림 1. 슈퍼프레임 구조
Fig. 1. Superframe Structure.

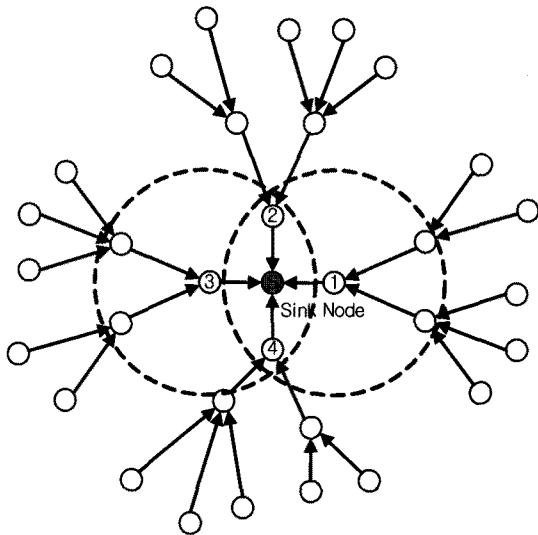


그림 2. 클러스터-트리 네트워크 예
Fig. 2. Example of Cluster-tree network.

번의 CCA실패가 일어나면 해당 패킷의 전송을 포기한다. CSMA-CA방식을 이용하면 적은 전력소비로 무선 충돌을 줄일 수 있다. 하지만 전파 도달거리 밖에 있는 노드들이 동시에 패킷을 전송하여 발생하는 히든 터미널 충돌과, 동일한 슬롯에서 CCA를 시작하여 동일한 슬롯에서 패킷을 전송하여 발생하는 충돌을 해결하지 못한다.

그림 2는 클러스터-트리 형태로 구성된 네트워크 예를 보여준다. IEEE 802.15.4는 최대 65535개의 노드를 네트워크에 참여 시킬 수 있다. 네트워크에 참여하는 노드가 많고 넓은 지역에 걸쳐서 형성되는 네트워크를 구성하기 위해서는 그림 2처럼 클러스터-트리 형태의 네트워크를 구성하는 것이 효과적이다.

클러스터-트리 형태의 네트워크는 그림 2에서 가운데 위치한 노드 S처럼 네트워크 전체의 코디네이터인 PAN 코디네이터, 1, 2, 3, 4번 노드처럼 상위 그룹에 참여하는 자식노드이면서 동시에 각각이 자신의 그룹에서 코디네이터로 동작하여 그룹과 그룹 사이에 데이터를 전달하는 라우팅노드와 네트워크의 가장자리에 위치한 종단노드들로 이루어진다.

그림 3은 그림 2와 같은 클러스터-트리 네트워크에서 1, 2, 3, 4번 노드와 같은 라우팅노드의 타이밍을 보여준다. 예를 들어 3번 노드는 자신의 그룹의 코디네이터로서 자신의 그룹에 비콘을 전송하여 Outgoing 슈퍼프레임을 시작한다. Outgoing 슈퍼프레임이 끝난 다음 싱크노드의 자식노드로서 싱크로부터 비콘을 수신하고 Incoming 슈퍼프레임에 참여하여 자신이 코디네이터로

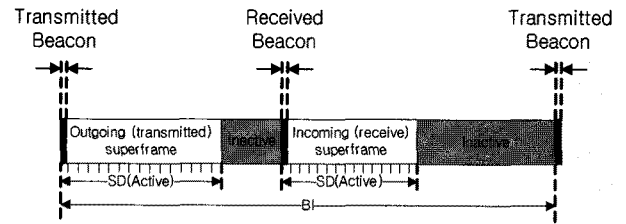


그림 3. 클러스터-트리 네트워크의 라우터 노드 타이밍
Fig. 3. Router node timing in cluster-tree topology.

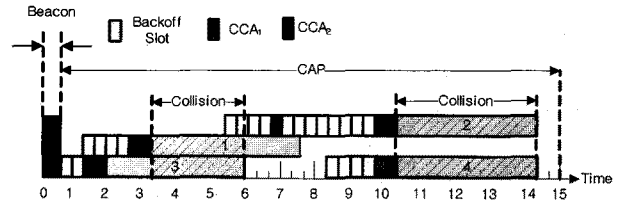


그림 4. 충돌발생 상황
Fig. 4. Situation when Collision occurred.

동작하는 그룹과 상위 그룹 사이에 데이터를 전달할 수 있다. 노드 1, 2, 3, 4는 모두 그림 3과 같은 타이밍을 가진다.

그림 4는 그림 2와 같은 클러스터-트리 네트워크에서 PAN 코디네이터로 네트워크의 정보가 전달되는 상황에서 싱크와 1홉 거리인 1, 2, 3, 4번 노드가 싱크로 데이터를 전송할 경우 발생하는 충돌 상황을 나타내었다. 그림 4에서 백오프 슬롯은 CSMA-CA에서 무선 충돌확률을 줄이기 위해서 CCA를 수행하기 전에 랜덤하게 선택하여 기다리는 구간이고, CCA₁은 백오프 슬롯이 끝나고 채널의 상태를 확인하는 구간이며 CCA₂는 CCA₁이 성공하면 다음 슬롯에 채널의 상태를 다시 확인하는 구간이다. 두 번의 CCA가 성공해야 노드는 패킷 전송을 시작한다.

그림 4의 상황을 자세히 설명하면 3번 노드는 랜덤 백오프 슬롯으로 2를 선택하고 2개의 슬롯을 기다린 후 CCA를 수행한다. 다른 노드들이 채널을 사용하지 않기 때문에 두 번의 CCA가 성공하여 패킷전송을 시작한다. 1번 노드는 랜덤 백오프 슬롯을 4로 결정하고 4개의 슬롯 기다린 후 CCA를 수행한다. 노드 1이 CCA를 수행할 때 채널은 3번 노드가 사용하고 있지만 1번 노드와 3번 노드는 서로 전파도달범위 밖에 있기 때문에 1번 노드는 두 번의 CCA가 성공하고 패킷 전송을 시작한다. 코디네이터에서는 1번 노드와 3번 노드의 패킷이 충돌했기 때문에 코디네이터는 수신한 패킷에서 CRC 에러를 발견하고 수신한 패킷을 버린다. 2번 노드는 백오프 슬롯으로 4를 선택하여 4개의 슬롯을

기다린 후 CCA를 수행한다. 2번 노드가 CCA를 수행할 때 채널은 1번 노드가 사용하고 있고 2번 노드는 1번 노드와 3번 노드 모두와 전파 도달거리 안에 있기 때문에 2번 노드의 CCA는 실패한다. 따라서 2번 노드의 백오프 슬롯은 더블링(doubling) 되어 8이 되고, 8개의 슬롯을 기다린 후에 CCA를 다시 수행하고, CCA를 수행할 때 채널을 사용하는 노드가 없기 때문에 두 번의 CCA가 모두 성공하게 되고, CCA₂ 다음 슬롯에서 패킷 전송을 시작한다. 4번 노드는 백오프 슬롯으로 4를 선택하고 4개의 슬롯을 기다린 후 CCA를 수행한다. 4번 노드의 CCA도 모두 성공하기 때문에 2번 노드와 4번 노드는 동일한 슬롯에서 데이터를 전송하기 시작하여 충돌이 발생한다.

많은 노드들이 참여하고, 넓은 지역에 걸쳐서 생성되는 클러스터-트리 구조의 네트워크가 모니터링 목적으로 사용된다면 네트워크에서 생성된 정보는 싱크로 전달되기 때문에 싱크에 가까울수록 트래픽이 집중된다. 이러한 싱크 주변 지역과 같이 트래픽이 많은 경우 그림 4와 같이 히든 터미널 충돌과, 동일한 슬롯에서 CCA를 시작해서 발생하는 충돌이 자주 생기고 CCA 충돌로 백오프 슬롯이 길어져 데이터의 전송지연이 커진다. 만약 CCA 실패로 길어진 백오프 슬롯이 현재 남은 액티브 구간보다 길면 패킷 전송은 다음 슈퍼프레임에 시도되므로 심각한 전송지연이 발생한다. 또한 CCA 실패가 macMaxCSMABackoffs 이상 되면 노드는 해당 패킷의 전송을 포기하게 된다. macMaxCSMABackoffs는 0-5까지의 값을 가질 수 있고 기본값은 4이다. IEEE 802.15.4에서는 이러한 상황을 해결하기 위해서 긴 CAP 구간을 사용하여 CSMA-CA가 실패할 확률을 낮출 수밖에 없다.

III. 동적 채널 할당 알고리즘

CAP구간은 노드들이 활동 상태에 있어야 하기 때문에 전력소비가 많고, 백오프 슬롯으로 인한 전송지연과 충돌로 인한 재전송 등의 오버헤드가 존재한다. 반면에 CFP구간은 폴링구간으로 전송지연이나 충돌이 없다. CFP구간은 IEEE 802.15.4 표준에서 전송지연에 민감한 어플리케이션에 GTS를 할당 하는데 사용된다. GTS는 CAP구간에서의 MAC 명령프레임을 이용하여 요청되고, GTS를 요청받은 코디네이터는 남은 자원이 있는 경우 다음 사이클의 슈퍼프레임부터 GTS를 할당한다.

GTS는 GTS 해제 요청을 받거나, 사용시간이 만료되거나 또는 일정시간 동안 사용되지 않으면 해제되고 해제되기 전까지는 매 슈퍼프레임마다 GTS가 할당되기 때문에 주기적으로 매번 존재하는 트래픽에 적절하며, 비주기적인 트래픽에 GTS를 할당하는 것은 비효율적이다.

센서 네트워크에서는 대부분의 데이터가 비주기적이기 때문에 데이터 전송에 CAP구간이 이용된다. 따라서 CAP구간의 길이가 적절해야 불필요한 에너지 낭비가 없다. 만약 트래픽에 비해서 CAP구간의 길이가 짧다면 노드들의 CCA가 실패할 확률이 높아져 전송지연이 길어지고, 전송시도 횟수 초과나 충돌로 인한 데이터 손실도 많아진다. 반면 트래픽에 비하여 CAP구간이 길면 데이터 전송은 원활해지지만 긴 활동구간을 유지해야 하므로 전력소비가 많아진다.

본 장에서는 불필요한 전력소비를 줄이고, 데이터 전송을 원활하게 하기 위하여 트래픽에 따라 채널을 동적으로 할당하는 알고리즘인 DCA(Dynamic Channel Allocation)를 제안한다. DCA는 GTS를 할당하는 알고리즘과, 네트워크의 트래픽을 예측하고 액티브구간의 길이를 조절하여 불필요한 전력소비를 줄이는 알고리즘으로 구성된다.

1. GTS할당 알고리즘

DCA는 IEEE 802.15.4의 슈퍼프레임 구조를 수정한 RSS(RTS Superframe Structure)를 사용하여 동적으로 GTS를 할당하고, CAP와 RTS 구간의 길이를 조절한다. 그림 5는 DCA가 사용하는 RSS구조를 보여준다.

RSS는 IEEE 802.15.4의 슈퍼프레임 앞에 RTS구간

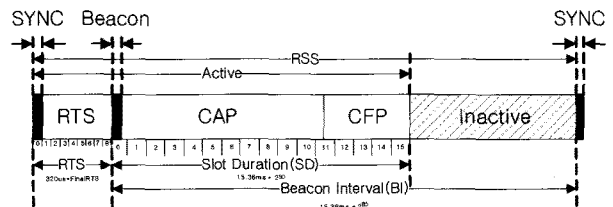


그림 5. 제안된 RSS 구조
Fig. 5. Proposed RSS structure.

Octets:	4	1	1	2	1	2	1	2
Preamble	SFD	Length	Frame Control	Sequence Number	PAN ID	Final RTS	FCS	
	SHR	PHR	MHR			MAC Payload		MFR

그림 6. SYNC프레임 구조
Fig. 6. SYNC frame structure.

을 삽입한 구조로서, RTS/CTS방식을 응용하여 GTS를 할당한다. RSS는 SYNC프레임을 수신하면서 시작하고, RTS구간에서는 GTS예약을 위해 RTS프레임이 사용된다.

SYNC프레임은 RTS구간을 정의하며 SYNC Header 5 바이트, PHY Header 1 바이트, MAC Header 5 바이트, MAC Payload 1바이트 MAC Footer 1 바이트로 전체 14바이트 길이를 가진다. SYNC프레임이 전송되면서 RTS구간이 시작되고, 수신한 SYNC프레임의 MAC Payload에 RTS구간의 끝이 정의되어있다. 그림 6은 SYNC프레임의 구조를 나타낸다.

RTS프레임은 14 바이트 길이로 SYNC Header 5 바이트, PHY Header 1 바이트, MAC Header 6 바이트, MAC Footer 2 바이트로 구성된다. MAC Header의 FCF(Frame Control Field)의 예약된 비트인 7-9번 비트에 데이터 전송에 필요한 슬롯길이를 포함시켜 전송한다. GTS Slot Request는 3비트 길이이므로 0-8까지 표현 가능하다. 예를 들어 데이터 전송에 3개의 슬롯이 필요하면 FCF의 7-9번 비트인 GTS Slot Request 필드에 3을 써서 RTS프레임을 전송한다. 그림 7은 RTS프레임의 구조를 나타내고, 그림 8은 수정된 FCF 구조를 나타낸다.

그림 9는 RTS프레임을 수신했을 때 CRC검사를 하여 충돌을 카운트하는 알고리즘이다. RTSCollisionCnt는 이번 RTS구간에서의 RTS프레임 충돌 횟수 카운터이다. CollisionCnt는 RTS프레임의 연속충돌 횟수 카운터이고, NoCollisionCnt는 RTS프레임이 연속으로 충돌하지 않은 횟수에 대한 카운터이다.

RTSCollisionCnt는 CAP에 추가적인 슬롯 할당에 사용되고, CollisionCnt와 NoCollisionCnt는 기본적인

Octets: 4	1	1	2	2	2	2
Preamble	SFD	Length	Frame Control	PAN ID	Src ID	FCF
SHR		PHR	MHR			MFR

그림 7. RTS프레임 구조
Fig. 7. RTS frame structure.

Bits: 0-2	3	4	5	6	7-9	10-11	12-13	14-15
Frame Type	Security Enabled	Frame Pending	Ack. Request	PAN ID Compression	GTS Slot Request	Dest. Addressing Mode	Frame Version	Source Addressing Mode

그림 8. 수정된 FCF프레임 구조
Fig. 8. Modified FCF frame structure.

```

RTSCollisionCheck(){
1  if(RTS_CRC_Error)then
2      RTSCollisionCnt = RTSCollisionCnt + 1
3      CollisionCnt = CollisionCnt + 1
4      NoCollisionCnt = 0
5  else
6      CollisionCnt = 0
7      NoCollisionCnt = NoCollisionCnt + 1
endif
}
    
```

그림 9. RTS구간 길이 조절 알고리즘
Fig. 9. Algorithm to dynamically adjusts the length of RTS.

CAP구간의 길이와 RTS구간의 길이를 조절할 때 사용

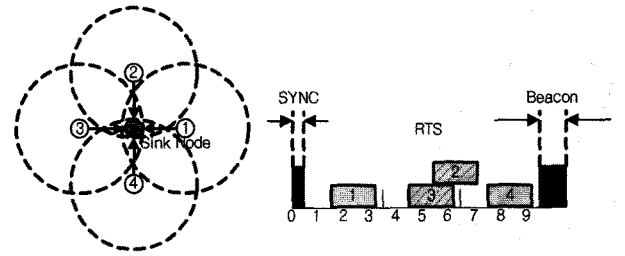


그림 10. RTS 전송 예
Fig. 10. Example of RTS transmission.

된다. 코디네이터로 전송할 데이터가 있는 노드는 GTS를 요청하기 위해 RTS구간에서 데이터 전송에 필요한 슬롯을 RTS프레임에 포함시켜 코디네이터로 전송한다. RTS프레임 전송은 랜덤한 슬롯에서 충돌방지 알고리즘 없이 전송된다.

그림 10은 그림 2의 상황에서 RTS구간의 동작을 보여준다. 노드 1, 2, 3, 4는 각각 RTS구간에서 랜덤한 슬롯에 RTS프레임을 전송한다. 노드 1번과 노드 4번은 RTS프레임이 충돌하지 않았지만 노드 2번과 노드 3번은 RTS프레임이 충돌했다.

코디네이터는 RTS프레임을 수신하면 곧바로 RTSCollisionCheck()를 수행한다. 수신한 RTS프레임에서 CRC에러를 발견한다면 RTSCollisionCnt와 CollisionCnt를 1 씩 증가시키고, NocollisionCnt는 0으로 리셋한 후 수신한 RTS프레임은 버린다. CRC에러가 발생하지 않았을 경우 NoCollisionCnt를 1 증가시키고, CollisionCnt를 리셋한 후 GTS를 할당한다. RTS구간이 끝날 때까지 코디네이터는 RTS프레임을 수신하고 수

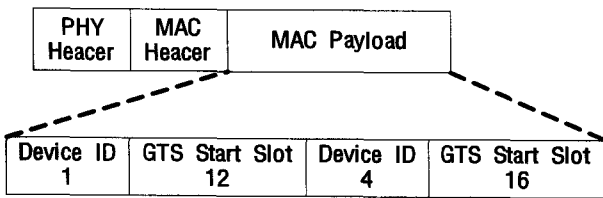


그림 11. GTS정보가 포함된 비콘
Fig. 11. Beacon which includes GTS information.

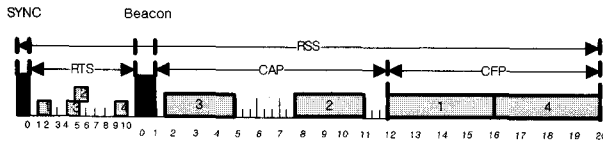


그림 12. RSS의 타이밍
Fig. 12. Timing of RSS.

신한 RTS프레임에 대하여 RTSCollisionCheck()를 수행하여 GTS를 할당한다. RTS구간이 끝나면 할당된 GTS들을 비콘에 삽입하여 브로드캐스팅 시킨다. 그림 11은 그림 10의 상황에서 RTS구간이 끝난 후 브로드캐스팅되는 비콘의 구조를 보여준다.

노드들은 RTS구간이 끝나면 브로드캐스팅되는 비콘을 수신한다. 수신한 비콘에 자신의 GTS가 할당되어 있으면 CFP구간에 자신에게 할당된 GTS를 이용하여 데이터를 전송하고, GTS를 할당받지 못했다면 CAP구간에서 CSMA-CA를 사용해 데이터 전송을 시도한다. 그림 11에서 나타난 GTS정보처럼 1번 노드는 12번 슬롯에서 데이터 전송을 시작하고, 4번 노드는 16번 슬롯에서 데이터 전송을 시작한다. 2번과 3번 노드는 GTS를 할당받지 못했으므로 CAP구간에서 데이터를 전송한다. 그림 12는 그림 10과 같은 상황에서 RSS의 동작 타이밍을 보여준다.

코디네이터는 CRC검사를 통해 RTS프레임 충돌을 인식했기 때문에 최소 두 개 이상의 노드가 CAP구간을 이용하려 한다고 판단할 수 있다. 따라서 CAP구간을 이용하는 노드들의 데이터 전송을 보장하기 위해서 CAP구간에 추가적인 슬롯을 할당한다. 그림 12에 나타난 CAP구간의 길이는 12개 슬롯인데, 이는 CAP구간의 길이를 8개 슬롯으로 유지하다가 RTS구간에서 하나의 충돌을 감지했기 때문에 CAP구간에 4개의 슬롯을 추가해서 12개 슬롯이 된 것이다. CAP구간의 길이를 조절하는 알고리즘은 다음 절에서 자세히 설명한다.

2. CAP구간의 길이 조절 알고리즘

CAP구간은 네트워크의 가입과 비콘 상실프레임 등

의 MAC 명령프레임들이 전송되는 구간이다. 또한 CAP구간은 RTS프레임 충돌로 인하여 GTS를 할당받지 못한 노드들이 데이터 전송을 시도하는 구간이므로 CAP구간을 이용하는 트래픽은 가변적이다. DCA는 CAP구간에서의 트래픽을 예측하여 적절하게 CAP구간의 길이를 조절함으로써 불필요한 전력소비를 줄인다.

그림 13은 동적으로 CAP구간 길이를 조절하는 알고리즘이다. 코디네이터는 RTS프레임의 충돌정보에 기초하여 기본적인 CAP구간 길이인 BaseCAP를 결정하고, RTS구간에서의 충돌정보를 바탕으로 추가적인 슬롯을 할당하여 이번 슈퍼프레임의 CAP 길이인 FinalCAP를 결정한다. RTS구간에서 충돌이 발생했다면 최소 두 개 이상의 노드가 CAP구간을 이용하게 될 것이므로 하나의 충돌당 4개의 슬롯을 추가적으로 할당한다. 만약 RTS구간에서 충돌이 없다면 CAP구간을 이용하는 트래픽이 많지 않다고 판단하고 추가적인 슬롯을 할당하지 않는다.

그림 10의 상황에서 2번과 3번의 RTS프레임이 충돌했기 때문에 RTSCollisionCnt가 1이 된다. 따라서 코디네이터는 현재의 BaseCAP인 8개의 슬롯에 4개의 슬롯

```

Dynamic_CAPLength(CollisionCnt, NoCollisionCnt,
1 BaseCAP, FinalCAP, ChildNum){
2 if ( CollisionCnt > 2 ) then
3     BaseCAP = BaseCAP + 1
4     CollisionCnt=0
5     if BaseCAP > ChildNum*3 then
6         BaseCAP = ChildNum*3
7     endif
8 endif
9 if ( NoCollisionCnt > 2 ) then
10    BaseCAP = BaseCAP - 1
11    NoCollisionCnt=0
12    if ( BaseCAP < 8 ) then
13        BaseCAP = 8
14    endif
15 endif
16 FinalCAP=BaseCAP + ( 4*RTSCollision )
RTSCollisionCnt=0 }
    
```

그림 13. 동적 CAP구간 길이 조절 알고리즘
Fig. 13. Algorithm to dynamically adjusts the length of CAP.

을 더하여 총 12개의 슬롯을 FinalCAP로 결정한다. FinalCAP를 결정하고 나면 RTSCollision은 0으로 초기화 해준다.

BaseCAP길이는 세 번 연속 RTS충돌이 발생하여 CollisionCnt가 3 이상이면 1만큼 증가하고, 세 번 연속 RTS충돌이 없어서 NoCollision이 3 이상이면 1만큼 감소한다.

BaseCAP길이의 최소값은 8개 슬롯으로 IEEE 802.15.4에서 정의된 CAP구간의 최소 길이인 440Symbol보다 40Symbol이 긴 480Symbol이다. BaseCAP길이의 최대값은 자식노드 수의 3배이다. CAP 구간 길이 조절 알고리즘에서는 aBaseSlotDuration을 기본 단위로 사용한다. aBaseSlotDuration은 60Symbol 길이이다. 1 Symbol은 16us이므로 aBaseSlotDuration은 960us이고 이는 2.4GHz대역에서 30바이트를 전송할 수 있는 길이이다.

3. RTS구간의 길이 조절 알고리즘

그림 14는 RTS구간의 길이 조절 알고리즘을 나타낸다. RTS프레임을 송신하는 노드들이 많아지면 RTS구간의 길이를 조절해 주어야 RTS프레임 충돌 횟수가 줄어들어 GTS이용률이 높다. RTS구간의 길이는 RTS프

```

Dynamic_RTSLength(Final_RTSLength, CollisionCnt,
1 NoCollision, ChildNum){
2   if ( CollisionCnt > 2 ) then
3     FinalRTS = FinalRTS + 2
4     CollisionCnt = 0
5     if ( FinalRTS > (ChildNum*4) ) then
6       FinalRTS = ChildNum*4
7     endif
8   endif
9   if ( NoCollisionCnt > 2 ) then
10    FinalRTS = FinalRTS - 2
11    NoCollisionCnt = 0
12    if ( FinalRTS < 8 ) then
13      FinalRTS = 8
14    endif
15  endif )

```

그림 14. RTS구간 길이 조절 알고리즘

Fig. 14. Algorithm to dynamically adjusts the length of RTS.

레이의 충돌정보를 이용하여 조절한다. CollisionCnt가 3 이상이면 트래픽이 많은 상황이라 판단하여 RTS구간 길이를 2 슬롯 증가시키고, 3번 연속 충돌이 없어 NoCollisionCnt가 3 이상이면 트래픽이 적은 상황이라 판단하고 RTS구간의 길이를 2 슬롯 감소시킨다. RTS구간 길이의 최소값은 8 슬롯이다. RTS구간길이의 최대값은 자식노드들의 숫자의 4 배 이다. RTS조절에 사용하는 슬롯은 aUnitBackoffPeriod단위를 사용하고 aUnitBackoffPeriod는 20Symbol로 2.4GHz대역에서 10바이트를 전송할 수 있는 길이이다.

IV. 구현 및 성능 분석

DCA의 성능을 평가하기 위해 IEEE 802.15.4와 비교 실험을 했다. 실험 환경은 멀티 홉으로 이루어진 클러스터-트리 토폴로지에서 네트워크에서 생성된 데이터는 싱크로 전달되는 상황을 가정했다. 이러한 상황에서 싱크와 1홉 거리의 지역에서 네트워크의 트래픽이 집중되므로, 싱크와 싱크의 1홉 거리에 위치하는 노드들의 트래픽을 분석하여 성능을 평가했다. 성능 평가는 노드의 액티브 구간의 길이, 데이터의 전송 성공률과, 채널 사용률을 기준으로 이루어졌다.

DCA는 네트워크의 트래픽에 따라서 동적으로 액티브 구간이 할당된다. 하지만 IEEE 802.15.4는 동적으로 액티브 구간을 조절하는 메커니즘이 없기 때문에, DCA와의 성능비교를 위해서 다양한 길이의 액티브 구간을 설정하여 실험했고, DCA와 비슷한 패킷 전송 성공률을 보이는 액티브 구간의 결과를 가지고 DCA와 비교했다.

싱크와 1홉 거리에 존재하는 노드들이 3, 5, 7개가 있는 경우를 각각 실험했고, 각각의 경우마다 싱크로 전달해야 하는 데이터가 있을 확률을 10-100%까지 변화시켜가며 실험했다. 각 상황별로 10000번의 반복실험을 통해 얻은 데이터의 평균을 사용하여 성능분석에 사용했다.

실험에 사용한 데이터패킷은 120바이트, SYNC패킷은 14바이트, RTS패킷은 14바이트 길이이고, 비콘 패킷의

표 1. 실험에 사용한 패킷 길이

Table 1. Packet length which is used for experiment.

	SYNC	RTS	Beacon	Data	Ack
Length	14	14	16+3*n	120	11
Slot	2	2	variable	12	2

표 2. CC2420의 전기적 특성

Table 2. Electronic features of CC2420.

	RX	TX	IDLE	SLEEP
Current(mA)	19.7	17.4	0.426	0.020
Power(mW)	35.46	31.32	0.77	0.036

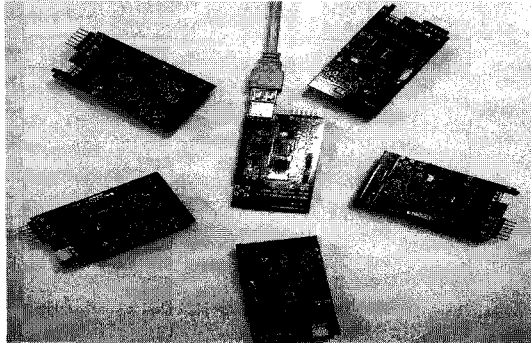


그림 15. 실험에 사용한 싱크와 싱크의 1홉 지역

Fig. 15. The sink node used in experiment and 1 hop from the sink.

길이는 가변적이다. 표 1은 실험에 사용한 패킷의 길이와 패킷 전송에 필요한 슬롯을 백오프 슬롯 단위로 나타낸 것이다.

DCA의 성능을 분석하기 위해서 Atmel사의 ATMEGA128과 Chipcon사의 CC2420로 구성된 하드웨어에 DCA를 구현하여 그림 15와 같은 구성으로 실험했다. 트래픽 정보를 얻기 위해서 인액티브 구간에서 노드마다 고유한 슬롯을 할당하였고, 노드들은 액티브 구간이 끝나면 액티브구간에서의 활동 정보를 자신의 고유한 슬롯에서 싱크로 전송한다. 싱크는 자신의 액티브 구간에서의 정보와 수신한 액티브구간에서의 정보를 UART통신으로 PC로 전송한다.

Physical Layer에 대한 설정으로 250kbps의 데이터 전송률을 갖는 2.4GHz 대역의 11번 채널을 사용했고 전파 도달거리는 10m로 제한했다. 그림 15는 실험에 사용한 싱크와 싱크의 1홉 지역을 보여주고 표 2는 CC2420의 소비전력을 보여준다.

1. 싱크와 1홉 거리에 3개의 노드가 존재하는 경우

싱크 주변에 3개의 노드가 존재하는 경우에 DCA의 성능을 평가하기 위해서, IEEE 802.15.4표준이 액티브 구간의 길이를 14.40ms, 19.20ms, 28.80ms를 가질 때와 비교분석 했다. Std는 IEEE 802.15.4를 의미하고 Std 뒤의 괄호 안의 숫자는 액티브 구간의 길이를 의미한다. 예를 들어 Std(14.40ms)는 14.40ms의 액티브 구간

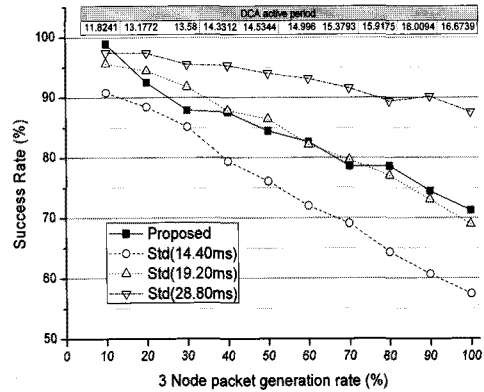


그림 16. 노드가 3개일 경우 성공률과 DCA 액티브 구간
Fig. 16. Success rate and active period of DCA when 3 nodes.

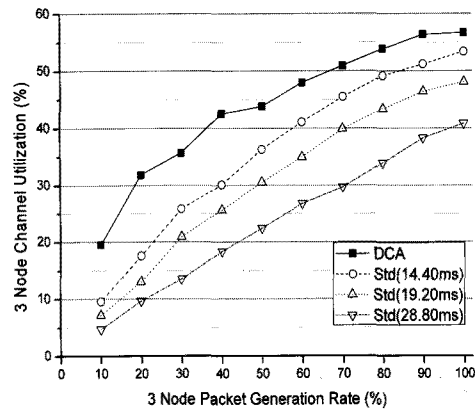


그림 17. 노드가 3개인 경우 채널 사용률
Fig. 17. Channel utilization when 3 node.

을 갖는 IEEE 802.15.4표준이다.

실험 환경은 싱크로 트래픽이 집중되는 상황이고, 패킷 생성률이 30%이하인 경우와 40%이상인 경우로 나누어서 분석했다.

그림 16은 액티브 구간 길이와 데이터 전송 성공률을 나타내는 그래프다. 그림 16에서 DCA active period는 각각의 데이터 생성률에서의 DCA가 갖는 액티브 구간 길이를 나타낸 것이다. 패킷 생성률이 30%이하일 때, DCA는 평균 12.86ms의 액티브길이를 갖고, 평균 93%의 전송 성공률을 보인다. Std(14.40ms)는 평균 88.1%의 성공률을 보이므로 DCA가 평균 10.7% 짧은 액티브 구간으로 평균 4.9%높은 전송 성공률을 보인다. 패킷 생성률이 40%이상인 경우는 DCA는 평균 15.41ms의 액티브 구간으로 평균 79.64%의 데이터 전송 성공률을 유지한다. Std(19.20ms)는 평균 79.29%의 데이터 전송

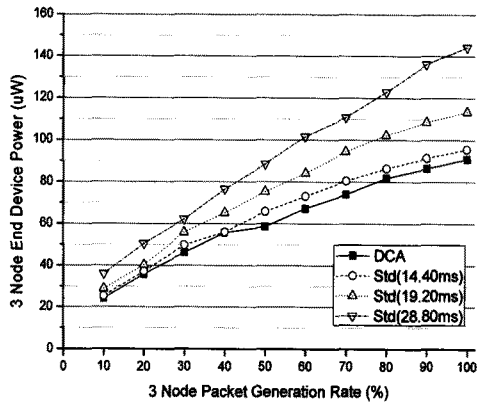


그림 18. 노드가 3개인 경우 노드들의 전력 소비량
Fig. 18. Power consumption of each node when 3 node.

성공률을 보이므로, DCA가 평균적으로 19.8% 짧은 액티브 구간을 유지하며 0.35% 높은 데이터 전송 성공률을 보인다.

그림 17은 채널 사용률을 비교한 그래프다. 패킷 생성률이 30% 이하일 경우 DCA는 평균 29%의 채널 사용률을 보이고, Std(14.40ms)는 평균 17.6%를 보인다. DCA가 Std(14.40ms)보다 평균 11.4% 높은 채널 사용률을 보인다. 패킷 생성률이 40% 이상인 구간에서는 DCA는 평균 50.3%의 채널 사용률을 보이고, Std(19.20ms)는 평균 38.41%의 채널 사용률을 보이므로 DCA보다 평균 12% 높은 채널 사용률을 보인다.

그림 18은 전력 사용량을 비교한 그래프다. 패킷 생성률이 30% 이하일 때 100번의 슈퍼프레임동안 DCA는 평균 35.3mW의 전력을 사용하고, Std(14.40ms)는 평균 37.35mW의 전력을 사용했기 때문에 DCA가 Std(14.40ms)보다 5.5%의 에너지를 절약한다. 패킷 생성률이 40% 이상인 경우 DCA는 73.5mW의 전력을 사용하고, Std(19.20ms)는 91.92mW의 전력을 사용했기 때문에 DCA가 Std(19.20ms)보다 20.3%의 에너지를 절약한다.

2. 싱크와 1홉 거리에 5개의 노드가 존재하는 경우 네트워크가 좀 더 큰 경우를 가정하여 싱크와 1홉 거리에 5개의 노드가 존재하는 경우에 성능을 비교분석했다. 트래픽이 늘어남에 따라서 IEEE 802.15.4가 DCA와 비슷한 데이터 전송 성공률을 보이기 위해 액티브 구간의 길이가 늘어났다. 이번 상황에서 DCA의 성능평가는 Std(19.20ms), Std(28.80ms), Std(38.40ms)와 비교하여 이루어졌다.

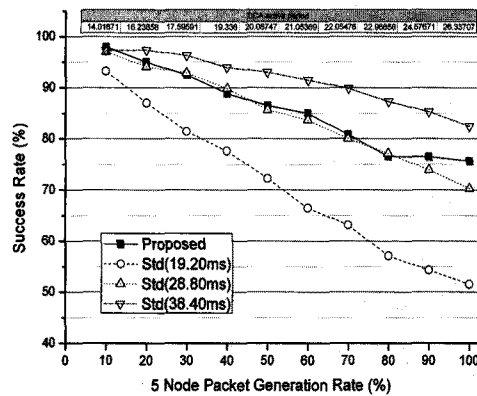


그림 19. 노드가 5개인 경우 전송 성공률과 DCA 액티브 구간
Fig. 19. Success rate and active period of DCA when 5 node.

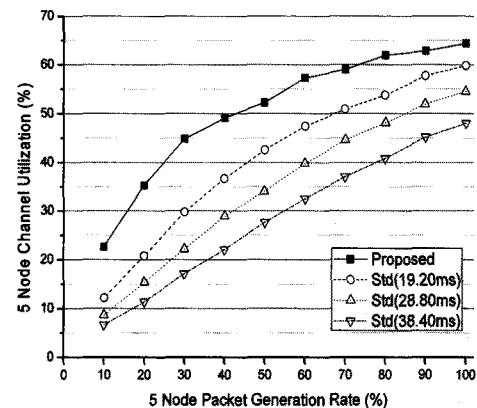


그림 20. 노드가 5개인 경우 채널 사용률
Fig. 20. Channel utilization when 5 node.

그림 19는 액티브 구간의 길이와 전송 성공률을 나타낸 그래프다. 패킷 생성률이 30% 이하일 때 DCA는 평균 15.9ms의 액티브 구간을 갖고 평균 95.1%의 전송 성공률을 보인다. Std(28.80ms)는 평균 94.7%의 전송 성공률을 보이므로 DCA가 평균적으로 45% 짧은 액티브 구간으로 0.4% 높은 전송 성공률을 보인다. 40% 이후의 구간에서 DCA는 평균 22.34ms의 액티브 구간을 가지고 평균 81.40%의 데이터 전송 성공률을 보인다. Std(28.80ms)는 평균 80.07%의 데이터 전송 성공률을 보이므로 DCA가 평균적으로 22.5% 짧은 액티브 구간으로 1.33% 높은 데이터 전송 성공률을 보인다.

그림 20은 채널 사용률을 나타낸 그래프다. 패킷 생성률이 30% 이하인 경우에 DCA는 평균 34.2%의 채널 사용률을 보이고, Std(28.80ms)는 평균 20.9%의 채널

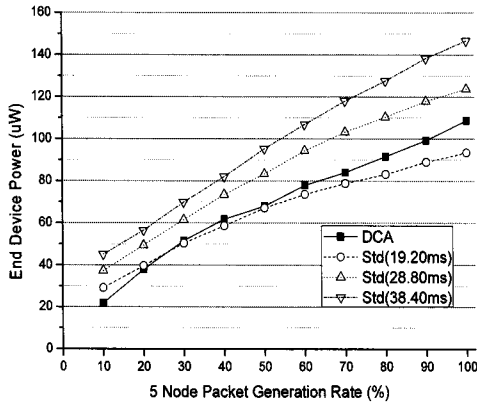


그림 21. 5노드일 경우 노드들의 전력 소비량
Fig. 21. Power consumption of each node when 5 node.

사용률을 보이므로 DCA가 평균 13.7% 높은 채널 사용률을 보인다. 패킷 생성률이 40% 이상인 경우 DCA는 평균 58.1%의 채널 사용률을 보이고, Std(28.80ms)는 평균 43.1%를 보인다. 따라서 DCA가 평균 15% 높은 채널 사용률을 보인다.

그림 21은 100번의 슈퍼프레임 동안 노드들의 전력 사용량에 대한 그래프이다. 패킷 생성률이 30% 이하인 경우 DCA는 평균 37.1mW의 전력을 사용했고, DCA와 비슷한 전송 성공률을 보이는 Std(28.80)는 평균 49.41mW의 전력을 사용했다. DCA가 Std(28.80)보다 평균적으로 25%의 에너지를 절약한다. 패킷 생성률이 40% 이상인 경우 DCA는 평균 84mW의 전력을 사용했고, Std(28.80ms)는 평균 100.9mW의 전력을 사용했기 때문에 DCA가 Std(28.80ms)보다 평균 16.1%의 전력을 절약한다.

3. 싱크와 1홉 거리에 7개 노드가 존재하는 경우

매우 큰 네트워크 구성으로 자식노드가 7개인 경우에 대한 실험이다. 이번 분석은 Std(28.80ms), Std(38.40ms), Std(48.00ms)의 결과와 DCA를 비교했다.

그림 22는 액티브 구간의 길이와 전송 성공률을 나타낸 그래프이다. 패킷 생성률이 30% 이하인 경우에 DCA는 평균 18.7ms의 액티브 구간의 길이에 평균 95.9%의 전송 성공률을 보이고, Std(38.40ms)는 평균 95.2%의 전송 성공률을 보인다. DCA가 Std(38.40ms)보다 평균적으로 51.8% 짧은 액티브 구간으로 0.7%의 높은 전송 성공률을 가진다. 40% 이상의 패킷 생성률에서 DCA는 평균 32.3ms의 활동구간으로 평균 86.6%의 전송 성공률을 가지고, Std(38.40ms)는 평균 80.8%의 전송 성공률을 가진다. DCA가 Std(38.40ms)보다 평균

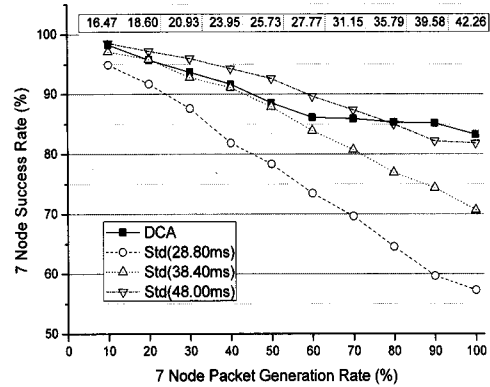


그림 22. 노드가 7개인 경우 전송 성공률과 DCA 액티브 구간
Fig. 22. Success rate and active period of DCA when 7 node.

사용률을 보이므로 DCA가 평균 13.7% 높은 채널 사용률을 보인다. 패킷 생성률이 40% 이상인 경우 DCA는 평균 58.1%의 채널 사용률을 보이고, Std(28.80ms)는 평균 43.1%를 보인다. 따라서 DCA가 평균 15% 높은 채널 사용률을 보인다.

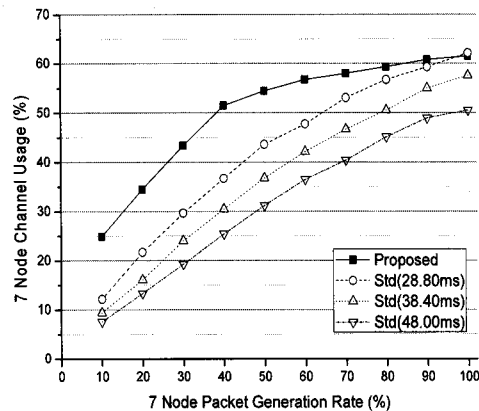


그림 23. 노드가 7개인 경우 채널 사용률
Fig. 23. Channel utilization when 7 node.

16%의 짧은 액티브 구간으로 5.8%의 높은 전송 성공률을 보인다.

그림 23은 채널 사용률을 나타낸 그래프이다. 패킷 생성률이 30% 이하인 경우에 DCA는 평균 34.2%의 채널 사용률을 보이고, Std(38.40ms)는 평균 18.4%의 채널 사용률을 보이므로 DCA가 평균적으로 16.5%만큼 채널 사용률이 높다. 패킷 생성률이 40% 이상인 경우 DCA는 평균 60.2%의 채널 사용률을 가지고, Std(38.40ms)는 평균 39.1%의 채널 이용률을 보이므로 DCA가 평균 15% 높은 채널 사용률을 가진다.

그림 24는 100번의 슈퍼프레임 동안 소비한 전력량을 비교한 그래프이다. 패킷 생성률이 30% 이하 구간에서 DCA는 평균 31.4mW의 전력을 사용했고, Std(38.40ms)는 평균 57.4mW의 전력을 사용했다. DCA는 Std(38.40ms)에 비해 평균 45.3%의 전력을 절약한다.

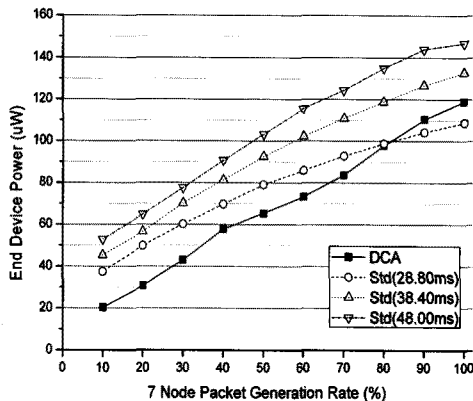


그림 24. 노드가 7개인 경우 노드들의 전력 소비량
Fig. 24. Power consumption of each node when 7 node.

패킷 생성률이 40%이상 구간에서 DCA는 평균 86.8mW의 전력을 사용했고, Std(38.40ms)는 평균 109.3mW의 전력을 사용했다. DCA가 Std(38.40ms)보다 평균 20%의 에너지를 절약한다.

데이터 생성률이 10%보다 낮을 경우에 IEEE 802.15.4는 14.40ms의 길이를 유지해야한다. 반면에 DCA는 최소 CAP구간 길이와 최소 RTS구간 길이인 9.6ms를 액티브 구간으로 유지하며 가끔씩 발생하는 데이터에 대해서만 추가적인 슬롯을 할당하므로, 데이터 생성률이 10% 이하인 상황에서 DCA는 IEEE 802.15.4보다 적은 전력을 사용하며 데이터 전송률이 10%인 경우의 데이터 전송 성공률인 98.9%보다 높은 전송 성공률을 보일 것이다.

V. 결 론

본 논문에서는 LR-WPAN에서 무선 충돌과 전송 지연을 줄이기 위해 트래픽에 따라 동적으로 채널을 할당하는 DCA를 제안했다. DCA는 RTS구간과 RTS프레임을 사용하는 오버헤드를 가지지만, DCA에서 사용하는 RTS구간의 길이는 CAP구간에 비해서 짧고, RTS프레임에 대한 GTS할당은 비콘에 삽입되어 한꺼번에 전송되므로 RTS/CTS사용에 따른 오버헤드 발생을 최소화한다. 또한 DCA는 트래픽을 예측하고 동적으로 채널을 할당한다. 따라서 갑작스러운 트래픽 집중상황에서도 노드의 전력 사용을 최소화하며 높은 데이터 전송 성공률을 유지할 수 있다.

데이터 생성률이 가장 낮을 환경에서, 가장 짧은 액

티브 구간을 갖는 IEEE 802.15.4와 DCA의 실험 결과를 비교했을 때 DCA가 더 적은 에너지를 사용하면서도 더 높은 데이터 전송 성공률을 보였다.

데이터 생성률이 실험에서 사용한 최소값보다 더 낮아진다면 DCA의 평균 액티브 구간이 짧아지지만, IEEE 802.15.4의 액티브 구간은 짧아질 수 없다. 따라서 DCA는 트래픽이 적은 상황에서도 IEEE 802.15.4보다 적은 에너지로 동작할 수 있다.

참 고 문 헌

- [1] 이상학, 조위덕 외 4명, "IEEE 802.15.4: Sensor Network 기술," *정보과학회지*, 제 21권, 제8호, 93-102쪽, 2003년 9월.
- [2] IEEE Standard for 802.15.4, "Part 15.4: Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Low-Rate Wireless Personal Area Networks(LR-WPANs)," 2006.
- [3] J. Zheng and M. J. Lee, "Will IEEE 802.15.4 make ubiquitous networking a reality? a discussion on a potential low power, low bit rate standard," *IEEE Commun.*, June 2004.
- [4] I. F. Akyildiz, Y. S.W. Su, and E. Cayirci, "A survey on sensor networks," *IEEE Commun.*, vol. 40, no. 8, pp. 102-114, Aug. 2002.
- [5] J. Mistic, S. Shafi, and V. B. Mistic, "The impact of MAC parameters on the performance of 802.15.4 PAN," *Ad Hoc Networks*, vol. 3, no. 5, pp.509-528, Sept. 2005.
- [6] Sofie Pollin, Mustafa Ergen, Sinem Coleri Ergen, Bruno Bougard, "Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer," *IEEE trans. Commun.*, vol. 7, no. 9, Sept. 2008.
- [7] L. Gang, B. Krishnamachari, and C. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *proc. IEEE Int. Conf. Performace, Compute., Commun.*, 2004, pp. 701-706.
- [8] T.O.KIM, H.Kim, J.Lee, J.S.Park, and B.D.Choi, "Performance analysys of the IEEE 802.15.4 with non beacon-enabled CSMA/CA in nonsaturated condition," in *proc. Int. Conf. EUC*, Aug. 2006, pp.884-893.
- [9] Jianliang Zheng and Myung J. Lee, "A comprehensive performance study of IEEE 802.15.4," *IEEE Press Book*, 2004.

[10] Shiann-Tsong Sheu, Yun-Yen Shin, "CSMA/CF Protocol for IEEE 802.15.4 WPANs," *IEEE trans. Commun.*, vol. 58, no. 3 Mar. 2009.

저 자 소 개



임 정 섭(학생회원)
2008년 인하대학교 전자공학과
학사 졸업.
2010년 인하대학교
전자공학과 석사 졸업.
2010년~현재 전자부품연구원
지능메카트로닉스 센터
위촉연구원

<주관심분야 : WSN, 임베디드 시스템, 컴퓨터
네트워크>



윤 완 오(학생회원)
2000년 경기대학교 전자공학과
학사 졸업.
2002년 인하대학교 전자공학과
석사 졸업.
2002년~현재 인하대학교
전자공학과 박사과정.

<주관심분야 : 분산 처리 시스템, 컴퓨터 구조, 임
베디드 시스템, 컴퓨터 네트워크>



서 장 원(학생회원)
2008년 인하대학교 전자공학과
학사 졸업.
2010년 인하대학교
전자공학과 석사 졸업.

<주관심분야 : 병렬 및 분산 처리 시스템, 컴퓨터
구조, 임베디드 시스템>



최 한 림(학생회원)
2008년 인하대학교 전자공학과
학사 졸업.
2010년 인하대학교
전자공학과 석사 졸업.
2010년~현재 삼성전자
반도체연구소 CAE팀

<주관심분야 : 컴퓨터 아키텍처, 병렬프로그래
밍>



최 상 방(평생회원)
1981년 한양대학교 전자공학과
학사 졸업.
1981년~1986년 LG 정보통신(주)
1988년 University of washington
석사 졸업.
1990년 University of washington
박사 졸업.

1991년~현재 인하대학교 전자공학과 교수
<주관심분야 : 컴퓨터 구조, 컴퓨터 네트워크, 무
선 통신, 병렬 및 분산 처리 시스템, Fault-
tolerant computing>