

논문 2010-47TC-9-4

# IEEE 802.15.4 LR-WPAN 환경에서 공정 큐잉을 위한 가중치 기반 GTS 할당 기법

( A Weight based GTS Allocation Scheme for Fair Queuing  
in IEEE 802.15.4 LR-WPAN )

이 경 화\*, 이 협 건\*\*, 신 용 태\*\*\*

( Kyoung-hwa Lee, Hyeopgeon Lee, and Yongtae Shin )

## 요 약

IEEE 802.15.4 표준에서는 QoS(Quality-of-Service)를 요구하는 응용프로그램들의 패킷 전송을 보장하기 위해 비경쟁 접근 방식에 의한 GTS(Guaranted Time Solot) 할당 기법을 지원한다. 그러나 표준에 따른 GTS 할당 기법은 패킷이 도착한 순서에 따라 채널을 할당하는 FIFS(First In First Service) 스케줄링을 사용해 패킷을 처리함으로써 대역의 낭비가 심하고 채널 사용률을 감소시켜 실질적인 QoS 서비스에 적합하지 못하다. 본 논문에서는 이러한 단점을 개선한 공정 큐잉을 위한 GTS 할당 기법을 제안한다. 제안하는 기법은 각 노드의 GTS 할당 요청 횟수(RC, Request Count)와 최근 할당 여부(RA, Recently Allocation)를 이용해 GTS 할당에 대한 불공정성 문제를 해결함으로써 GTS 요청 처리율을 높이고 지연 시간을 감소시킨다. 제안하는 기법의 타당성 및 성능 분석을 위해 수학적 분석 모델을 수립하고, 시뮬레이션을 통해 제안하는 기법이 표준과 비교해 데이터 처리율과 채널 사용률 측면에서 효율적임을 보였다.

## Abstract

The GTS(Guaranteed Time Slot) of the IEEE 802.15.4 standard, which is the contention free access mechanism, is used for low-latency applications or applications requiring specific data bandwidth. But it has some problems such as delay of service due to FIFS(First In First Service) scheduling. In this paper, we proposes a weight based GTS allocation scheme for fair queuing in IEEE 802.15.4 LR-WPAN. The proposed scheme uses a weight that formed by how much more weight we give to the recent history than to the older history for a new GTS allocation. This scheme reduces service delay time and also guarantees transmission simultaneously within a limited time. The results of the performance analysis shows that our approach improves the performance as compared to the native explicit allocation mechanism defined in the IEEE 802.15.4 standard.

**Keywords :** IEEE 802.15.4 LR-WPAN, Wireless Sensor Networks, GTS, Fair Queuing

## I. 서 론

<sup>[1]</sup>IEEE 802.15.4는 LR-WPAN(Low Rate Wireless

Personal Area Network)의 PHY와 MAC에 대한 표준으로 저속의 통신대역과 저전력을 목표로 하는 무선 기술이다. 802.15.4 표준은 에너지를 효율적으로 사용하기 위하여 비컨 프레임(beacon frame)에 의해 구분되는 슈퍼프레임(superframe) 단위로 동작하도록 규정한다. 하나의 슈퍼프레임은 활성(active) 구간과 비활성(inactive) 구간으로 나누어지며 활성 구간은 다시 경쟁 기반의 CAP(Contention Access Period) 구간과 비경쟁 기반의 CFP(Contention Free Period) 구간으로 나누어

\* 정회원, \*\* 학생회원, \*\*\* 정회원,  
숭실대학교 컴퓨터학과

(Department of Computingt, Soongsil University)

\* 본 연구는 지식경제부 및 정보통신연구진흥원의 IT 성장동력기술개발사업의 일환으로 수행되었습니다  
[2008-S-041-01, u-City용 센서네트워크 PHY/MAC 개발]  
접수일자: 2010년8월7일, 수정완료일: 2010년9월15일

진다. GTS(Guaranteed Time Slot)는 CFP 구간에서 QoS (Quality-of-Service)를 요구하는 응용프로그램들의 프레임 전송을 보장하기 위해 사용된다.<sup>[2-4, 7]</sup> 그러나 표준에서 제안하는 GTS 할당 기법은 FIFS(First In First Service) 스케줄링을 사용해 패킷을 처리함으로써 대역의 낭비가 심하고 채널 사용률을 감소시켜 실질적인 QoS 서비스에 적합하지 못하다.

이러한 문제점을 해결하기 위하여 본 논문에서는 공정 큐잉을 위한 가중치 기반 GTS 할당 기법을 제안한다. 제안하는 기법은 각 노드의 GTS 할당 요청 횟수(RC, Request Count)와 최근 할당 여부(RA, Recently Allocation)를 이용해 GTS 할당에 대한 불공정성 문제를 해결함으로써 지연 시간을 감소시키고 데이터 처리율을 높인다.

본 논문의 구성은 다음과 같다. II장에서는 IEEE 802.15.4의 MAC 구조 및 GTS 할당 기법의 문제점을 살펴본다. III장에서는 본 논문에서 제안하는 공정 큐잉을 위한 GTS 할당 기법을 제시한다. IV장에서는 제안한 기법의 성능을 분석하고 마지막 V장에서는 결론 및 향후 연구 과제를 제시한다.

## II. IEEE 802.15.4 MAC 구조

<sup>[1]</sup>IEEE 802.15.4는 비컨 사용 유무에 따라 비컨 모드(Beacon Enabled Mode)와 논 비컨 모드(None-Beacon Enabled Mode)로 동작한다. PAN 코디네이터는 디바이스와 통신을 위해 슈퍼프레임에 포함된 비컨을 사용해 동기화를 맞춘다.

### 1. 슈퍼프레임 구조

<sup>[1]</sup>IEEE 802.15.4는 에너지 효율적인 통신을 위해 슈퍼프레임 구조를 사용한다. 슈퍼프레임은 PAN 코디네이터라고 부르는 네트워크 코디네이터에 의해 일정 간격으로 비컨을 전송한다. 비컨 프레임은 각 슈퍼프레임의 첫 번째 슬롯에서 전송되며, 기기들을 동기화시키고 통신을 하는데 사용된다. 비컨 프레임 간격은 최소 15ms 에서 최대 245sec가 되며 첫 번째 비컨을 포함하여 다음 비컨 수신 직전의 시간까지로 하나의 *BI* (Beacon Interval)를 구성한다. (그림 1)은 이러한 슈퍼프레임 구조를 나타낸다.

슈퍼프레임은 *BI*를 기준으로 노드의 Active 구간(*SD*:Superframe Duration)과 Inactive 구간으로 나뉜

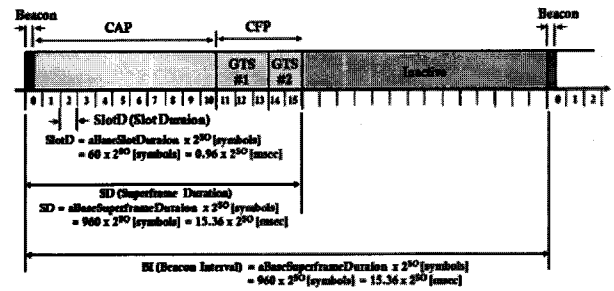


그림 1. 슈퍼프레임 구조의 예  
Fig. 1. An example of the superframe structure.

다. Active 구간은 동일한 크기의 16개 슬롯으로 나뉘며, CSMA/CA로 동작하는 CAP(Contention Access Period)와 GTS로 동작하는 CFP(Contention Free Period)로 구성된다. 통신을 원하는 노드는 CAP 구간에서 CSMA/CA를 사용한 경쟁을 통해 채널에 접근하고, CFP 구간에서 PAN 코디네이터로부터 GTS를 할당받아 경쟁 없이 채널에 접근한다.

슈퍼프레임 크기는 *PIB*(PAN information base) 파라미터값인 *BO*(Beacon Order)와 *SO*(Superframe Order)를 사용한 식 (1)에 의해 계산된다.

$$\begin{aligned} SD &= aBaseSuperframeDuration \cdot 2^{SO} \\ BI &= aBaseSuperframeDuration \cdot 2^{SO} \quad (1) \\ (0 \leq SO \leq BO \leq 14) \end{aligned}$$

*BO*는 Inactive 구간의 길이를 설정해 코디네이터가 비컨 프레임을 전송하는 주기를 알려준다. *SO*는 슈퍼프레임의 Active 구간의 길이를 설정하며 비컨 프레임을 포함한다.

### 2. CAP (Contention Access Period)

<sup>[1]</sup>IEEE 802.15.4 표준에 따르면 비컨모드 통신을 하는 모든 센서 노드들은 CAP 구간에서 채널에 접근하기 위해 CSMA/CA를 사용한다. 만약 어떤 노드가 채널 접근에 실패하게 되면, 해당 노드는 다음 슈퍼프레임의 CAP구간에서 채널이 유희상태(idle)일 때까지 기다려야 한다. (그림 2)는 CSMA/CA 알고리즘을 나타낸다.

CSMA/CA 알고리즘은 *NB*(Number of Backoff), *BE*(Backoff Exponent)와 *CW*(Contention Window)의 3가지 변수를 사용해 랜덤한 Backoff 구간을 사용함으로써 패킷의 충돌을 막는다. 전송할 데이터가 있는 노드는 채널이 사용 중인 경우  $random[0, (2^{BE} - 1)]$  동안 기다린 다음, *CW* 만큼의 CCA(Clear Channel Assessment)를 수행한다. 이때, Backoff 구간 설정을

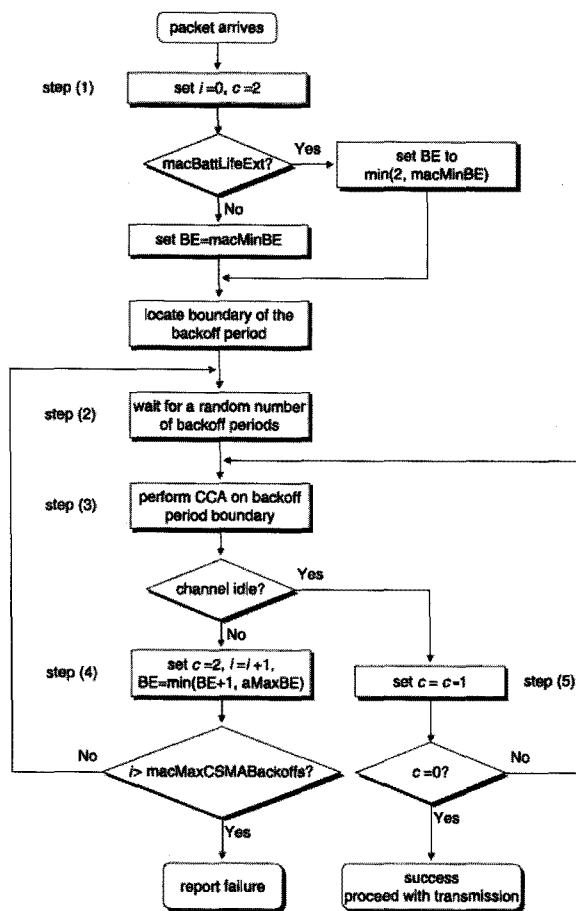


그림 2. CSMA/CA 알고리즘  
Fig. 2. The CSMA/CA algorithm.

위한 BE값을 작게 함으로서 빠른 요청 처리를 할 수 있다. 2회 CCA를 모두 성공하면 데이터를 전송하고 실패하면 NB 만큼의 backoff 단계와 CCA 단계를 반복한다. 하나의 Active 구간에서의 최소 CAP 심볼수 (aMinCAPLength)는 440 Symbols이다.

MAC은 PHY로부터 받은 데이터를 처리하기 위해 IFS(Inter-Frame-Space)라는 특정 타임 슬롯을 사용한다. IFS 는 데이터의 길이에 따라 SIFS(Short IFS)와 LIFS(Long IFS)로 나눈다. 전송하려는 데이터 프레임이 aMaxSIFSFrameSize(41symbols) 보다 작으면 SIFS (12symbols)를 그렇지 않으면 LIFS(40symbols)를 사용한다.

### 3. CFP (Contention Free Period)

[1, 7~8] GTS는 성형(star) 네트워크 토폴로지에서 코디네이터와 최대 255개 노드 사이의 QoS를 요구하는 프레임 전송을 보장하기 위해 사용된다. GTS를 할당받기 위해서 각 노드는 CSMA/CA 과정을 통해 PAN 코

디네이터에게 GTS 할당요청을 한다. 노드가 슈퍼프레임의 CAP 구간에서 경쟁을 통해 GTS 할당을 요청하면, 다음 비컨 신호에서 GTS 할당 확인이 이루어진다. GTS를 할당받은 노드는 비컨 프레임의 GTS Descriptor 필드에 명시된 GTS Starting Slot으로부터 GTS length 필드에 명시된 길이만큼 타임 슬롯을 독점적으로 사용한다. PAN 코디네이터는 슬롯 단위로 최대 7개까지 GTS를 할당한다.

## III. IEEE 802.15.4 GTS 성능 분석

### 1. 분석 모델

심볼률은 심볼수를 시간으로 변환하기 위해 필요하다. [1] 표준에 따르면, 2.45GHz 대역의 LR-WPAN에서 비트율은 250Kbps이며 심볼률은 62.5Ksymbols/sec이다. 하나의 심볼당 비트수는 O-QPSK(Offset Quadrature Phase-Shift Keying) 16 array 변조 방식을 사용하기 때문에  $2^{SO+2}$ 인 4bit이고, 하나의 심볼은 16μ sec 처리 속도를 갖는다.

$$bits\ per\ symbol = \frac{250Kbps}{62.5Ksymbols/sec} \quad (2)$$

(식 2)를 통해 symbols per octet은 2를 계산할 수 있다.

PAN 코디네이터와 노드 간에는 히든(hidden) 노드 문제가 없을 경우, BI동안 SD구간에서 전송할 수 있는 최대 심볼수는 비컨 프레임 심볼수와 CAP, CFP 구간의 심볼수의 총 합과 같다.

$$S_{SD} = S_{beacon} + S_{CAP} + S_{CFP} \quad (3)$$

CAP 구간에서 노드는 CSMA/CA를 통한 채널접근 권한을 얻기 위해 임의로 정해진 backoff 슬롯 개수만큼 지연 후, 채널이 유희 상태라면 2번의 CCA 과정을 수행한다. 노드에서 i번째 발생하는 backoff 수를  $NB_i$ 라고 한다면,

$$NB_i = random(2_i^{BE} - 1) \quad (4)$$

이고, 표준에 정의된 최대 backoff 시도 횟수 [1](macMaxCSMABackoffs)는 0에서 5까지의 값을 갖는다. macMaxCSMABackoffs값이 n인 경우 채널을 할당받는데 필요한 총 backoff 심볼수는 식 (5)와 같다.

$$L_{backoff} = 2 \left( \sum_{i=1}^n (NB_i) + 2T_{CCA} \right) \quad (5)$$

채널 접근 권한을 획득하면, 노드는 코디네이터에게 데이터 요청 프레임을 전송하고 이후 코디네이터로부터 응답 프레임을 받는다. 각 프레임에 대한 응답 시간은 식 (6)과 같다.

$$t_{ack} = aTurnaroundTime + aUnitBackoffPeriod \quad (6)$$

IFS 이후 코디네이터는 노드에게 데이터 프레임을 전송하고, 노드는 데이터 프레임의 FCS 결과 에러가 없으면, 이후에 ACK frame을 전송하여, 코디네이터에게 데이터 프레임을 성공적으로 수신하였음을 알린다. 하나의 비컨 주기 동안 평균 IFS 패킷 길이는 식 (7)과 같다.

$$L_{IFS} \approx \frac{L_{LIFS} \times macMinLIFS + L_{SIFS} \times macMinSIFS}{2} \quad (7)$$

$$IFS = (MPDU \leq aMaxSIFSFrameSize) ? \\ aMinSIFSPeriod : aMinLIFSPeriod$$

(식 3)~(식 7)를 통해 CAP 구간에서의 데이터 전송에 필요한 총 심볼수는 식 (8)과 같다.

$$S_{CAP} = L_{backoff} + L_{DataFrame} + L_{IFS} + L_{T_{ack}} + L_{AckFrame} + L_{IFS} \\ S_{CAP} = \begin{cases} T_{CAP} & (T_{CAP} > aMinCAPLength) \\ aMinCAPLength & (otherwise) \end{cases} \quad (8)$$

식 (8)의  $T_{CAP}$ 를 통해 superframe order를 계산하면 식 (9)와 같다.

$$SO = \left\lceil \log_2 \frac{T_{CAP}}{aBaseSuperframeDuration} \right\rceil \quad (9)$$

CFP 구간의 길이는 슈퍼프레임의 Active 구간에서 비컨 프레임과 CAP 구간의 길이를 제외한 길이와 같다. 따라서 하나의 비컨 주기 내에서의 CFP 구간의 길이는 식 (10)과 같다.

$$L_{CFP} = L_{Packet} - (L_{CAP} + L_{IFS} + L_{t_{ack}}) - at \quad (10)$$

여기서  $at$ 는 패킷을 송수신하는데 소요되는 시간을 나타낸다.

CAP 구간에서 가질 수 있는 최소 심볼 크기<sup>[1]</sup>  $aMinCAPLength$ 는 440이고, PAN 코디네이터는 슬롯 단위로 최대 7개까지 GTS를 할당할 수 있다. 따라서 CFP 슬롯의 최소 오프셋은 전체 SD 구간에서 비컨

구간과 CAP 구간을 제외한

$$CFPoffset = \left( 16 - Beacon_{slot} - \left\lfloor \frac{aMinCAPLength}{aBaseSlotDuration} \right\rfloor \right) \quad (11)$$

이며, 식 (11)을 통해 하나의 비컨 주기에서 CFP가 시작되는 최초 오프셋의 심볼값은

$$CFPoffset_{slot} = SD - Beacon_{slot} - (T_{CAP} \times 2^{SO}) \quad (12)$$

이다. 예를 들어, 슈퍼프레임이  $SO=2$ 이고 CAP가 최소 심볼 크기를 갖는다면,  $SD=3840$ ,  $T_{CAP}=1760$ symbols,  $CFPoffset_{slot}=1840$ symbols이 된다. 식 (2)~식 (12)를 통해 CFP 슬롯의 최소 오프셋은 9이고, 비컨 신호를 송신한 후 29.44ms 후에 CFP의 슬롯을 받을 수 있다.

## 2. GTS 처리율 분석

(그림 3)은 노드 수 증가에 따른 GTS 전체 처리율을 나타낸다.

$SO$ 값을 0에서 5까지 변화를 주면서 노드 수 증가에 따른 GTS 구간의 전체 심볼수를 측정하였다. 성능 분석 결과  $SO$ 값이 증가할수록 하나의 슬롯 당 심볼 크기가 증가하기 때문에 단위 시간 당 처리할 수 있는 패킷 수가 증가한다. 지연시간과 비교해보면  $SO$ 값이 증가할수록 하나의 슬롯 크기가 증가하기 때문에 데이터 처리율은 향상되지만 이에 따른 지연시간이 증가된다.

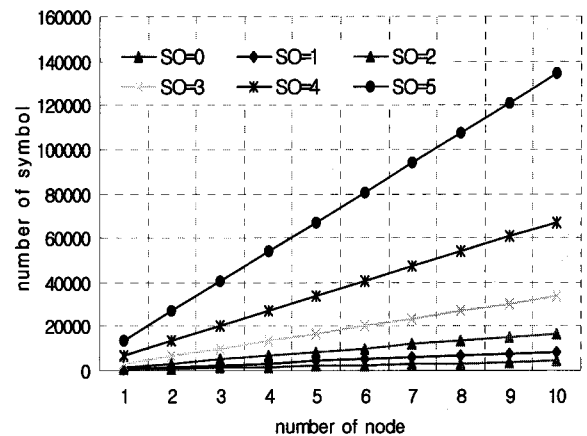


그림 3. 노드수 증가에 따른 GTS 전체 처리율

Fig. 3. The GTS throughput according to the number of nodes.

### 3. GTS 지연시간 분석

(그림 4)는 슈퍼프레임의 슬롯 크기를 결정하는 SO 값을 증가시켰을 경우 CAP와 GTS 각 구간에서 데이터 처리에 따른 지연 시간을 나타낸다. SO값이 커질수록 CAP 구간보다 GTS 구간에서의 패킷 처리 지연 시간이 줄어드는 것을 확인할 수 있다.

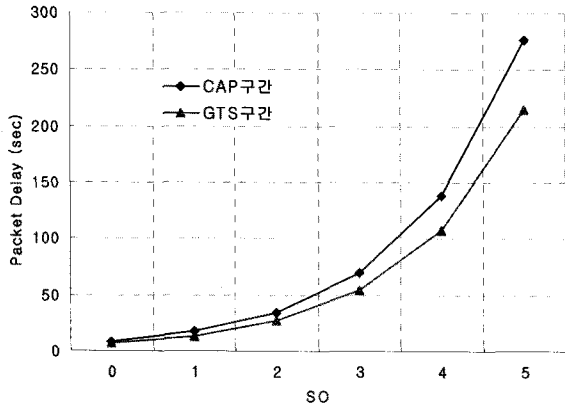


그림 4. 노드수 증가에 따른 GTS 지연시간  
Fig. 4. The Delay time of the GTS according to the number of nodes.

## IV. 제안하는 가중치 기반 GTS 할당 기법

<sup>[1]</sup>표준에서는 QoS(Quality-of-Service)를 요구하는 응용프로그램들의 패킷 전송을 보장하기 위해 비경쟁 접근 방식에 의한 GTS 할당 기법을 지원한다. 그러나 표준에 따른 GTS 할당 기법은 패킷이 도착한 순서에 따라 채널을 할당하는 FCFS 스케줄링을 사용함으로써 대역의 낭비가 심하고 채널 사용률을 감소시켜 실질적인 QoS 서비스에 적합하지 못하다.

이에 본 논문에서는 공정 큐잉을 위한 가중치 기반 GTS 할당 기법을 제안한다. 제안하는 기법은 각 노드의 GTS 할당 요청 횟수(RC, Request Count)와 최근 할당 여부(RA, Recently Allocation)를 이용해 가중치를 계산하고, 가중치별 우선 순위에 따라 GTS를 할당함으로써 GTS 요청 처리율을 높이고 지연 시간을 감소시킨다.

제안하는 기법은 다음 두 단계로 나뉘어 동작한다. 우선, “가중치 기반 GTS 요청 등급화 단계”에서는 RC 값과 RA 값을 이용해 가중치를 계산하고, GTS 요청 시 이 값을 PAN 코디네이터에게 전달한다. “가중치 우선순위에 따른 GTS 할당 단계”에서는 PAN 코디네이

터가 GTS 요청 패킷의 가중치 정보를 이용해 패킷의 우선순위를 분류하고 GTS를 할당해 노드와 통신을 수행한다.

### 1. 가중치 기반 GTS 요청 등급화 단계

성형 네트워크로 구성된 LR-WPAN 환경에서 코디네이터와 통신을 하기 위해  $N, (N=1,2,\dots,N)$ 개의 노드가 있다고 가정한다.  $U_i^r, (0 < i \leq N)$ 는  $r$ 번째 비컨 주기  $BL_r, (r=1,2,\dots,r)$ 에서 GTS 할당 요청을 하는 노드로 정의한다.  $U_i^r$ 가 GTS를 할당 받기 위해서 PAN 코디네이터에게 전송하는 패킷을  $P_i$ 라 정의하고, 각 패킷은 고정된 길이를 갖는다고 가정한다.  $P_i$ 는  $\lambda_i$ 를 발생 확률로 하는 Poisson 분포를 갖는다. 따라서  $U_i$ 에서  $BL_r$  동안 총 발생하는 GTS 할당 요청 확률은 식 (13)과 같다.

$$P(U_i) = e^{-\lambda} \frac{\lambda^r}{r!} \quad (\forall U_i | 0 < i \leq N, 0 < r) \quad (13)$$

$U_i^r$ 는 CSMA/CA 과정에서  $P(U_i)$  확률로 PAN 코디네이터에게  $P_i$ 를 전송한다. 이 때, 자신의 GTS 할당 요청 횟수(RC), 최근 GTS 할당 여부(RA)에 따른 가중치를 계산해 함께 전달한다. 노드  $U_i^r$ 가 GTS 할당 요청 시 사용할 가중치  $W_{U_i^r}$ 는 식 (14)와 같다.

$$W_{U_i^r} = \{RC_i^r, RA_i^r\} \quad (14)$$

식 (14)에서  $RC_i^r$ 는  $BL_r$ 에서 노드  $U_i^r$ 가 가진 패킷 전송을 위한 GTS 할당 요청 횟수를 나타낸다. 이전 비컨 주기  $BL_{r-n}, (n=1,\dots,n \text{ and } r > n)$ 에서 GTS 할당 요청을 하였으나, 현재 비컨 주기  $BL_r$ 에서 GTS를 할당받지 못한 경우, 요청 횟수에 따라  $RC_i^r$  값의 카운트를 1씩 증가시킨다. 만일,  $BL_r$ 에서  $RC_i^r > 0$ 인 경우, 처음 GTS 할당 요청을 한  $r-n$ 번째 비컨 주기로 부터 총  $n$ 번의 비컨 주기와 재요청이 발생했음을 의미한다.  $BL_r$ 에서 GTS 할당이 이루어지면  $RC_i^r$  값은 0으로 초기화 된다.

$RA_i^r$ 는  $BL_r$ 에서 노드  $U_i^r$ 의 GTS 할당 요청에 대한 GTS 할당 여부를 나타낸다. 이전 비컨 주기  $BL_{r-n}$ 에서 GTS 할당 요청을 하고, 현재 비컨 주기  $BL_r$ 에서 GTS를 할당받는 경우,  $RA_i^r$  값은 1이다. 즉,  $BL_r$ 에서  $RA_i^r$

**알고리즘 1. RC와 RA값을 이용한 가중치 계산**

```

1 //  $U_i (i=1...N)$  has  $GTS.request$ 
2  $RC_i^r = RA_i^r = 0;$ 
3  $BI_r = 1;$ 
4 for each node  $U_i$ 
5 //GTS를 할당받은 경우 RC, RA값 초기화
6 if( $GTS.allocated$ )
7      $RC_i^r = 0;$ 
8      $RA_i^r = 1;$ 
9 //GTS를 할당받지 못한 경우 RC 카운트 1 증가
10 else
11      $U_i^r \leftarrow RC_i^r + 1;$ 
12 //RC, RA 값을 이용한 가중치 값
13  $W_{U_i^r} = \{RC_i^r, RA_i^r\}$ 
14 //가중치 값을 포함해 GTS 할당 요청 재전송
15  $GTS.request(W_{U_i^r});$ 
16  $BI_r ++;$ 
17

```

그림 5. RC와 RA값을 이용한 가중치 계산 알고리즘  
Fig. 5. The weight calculation algorithm using RC and RA.

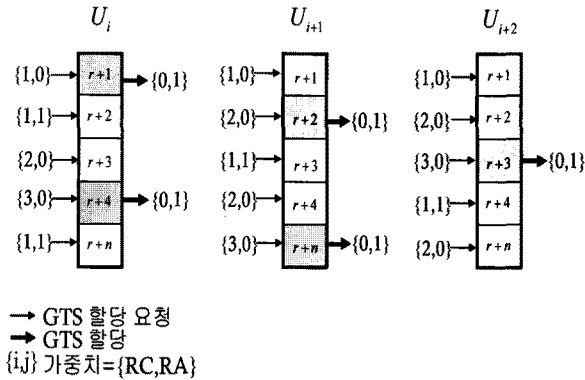


그림 6. 제안하는 알고리즘에 따른 RC와 RA값의 변화  
Fig. 6. Change the RC and RA using proposed algorithm.

값이 0이면, 이전 요청에 대한 GTS 할당이 이루어지지 않았음을 의미한다. 전송하려는 패킷에 대한 GTS 할당이 이루어지면  $RC_i^r$  값은 0으로,  $RA_i^r$  값은 1로 각각 초기화된다. (그림 5)는 RC와 RA 값을 이용한 GTS 요청 패킷의 가중치를 계산하는 알고리즘을 나타낸다.

(그림 6)은 임의의 세 개의 노드가 총  $n$ 번의 비컨 주기 동안 동시에 GTS 할당 요청을 하고 오직 1개의 노드만 GTS를 할당받는 경우, 제안하는 알고리즘에 따른 RC와 RA 값의 변화를 나타낸다.

노드  $U_i$ 가  $BI_r$ 에서 GTS 할당 요청을 할 경우, GTS 할당은 다음 비컨 주기인  $BI_{r+1}$ 에서 이루어진다. 이때, GTS 할당 요청에 대한 가중치  $W_{U_i^r}$ 는 {1,0} 값을 갖고, GTS 할당이 이루어지면  $W_{U_i^{r+1}}$ 는 {0,1}로 초기화된다.

다음 패킷 전송을 위한  $BI_{r+2}$ 에서  $U_i$ 가 GTS 할당 요청을 할 경우, 요청에 대한 가중치  $W_{U_i^{r+2}}$ 는 RC 카운트를 1 증가시킨 {1,1} 값을 갖는다. 그러나 GTS 할당을 받지 못해  $BI_{r+3}$ 에서 GTS 할당 재요청이 이루어질 경우,  $W_{U_i^{r+3}}$  값은 카운트를 1 증가시킨 RC값과 이전 요청에 대한 GTS 할당이 이루어지지 않은 RA값 0을 조합 한 {2,0}을 갖는다. 이러한 방식을 통해  $BI_{r+n}$ 까지  $W_{U_i^{r+n}}$  계산이 이루어진다.

**2. 가중치 우선순위에 따른 GTS 할당 단계**

GTS 할당 요청을 받은 PAN 코디네이터는 가용한 GTS 슬롯을 검사한다. 만약 가용한 GTS 슬롯 수보다 요청 수가 많을 경우, 먼저  $W_{U_i^r}$  값을 비교해 가장 큰  $W_{U_i^r}$  값을 갖는 노드에게 GTS를 할당한다. 동일한  $W_{U_i^r}$  값을 갖는 노드가 2개 이상 존재할 경우, 랜덤하게 GTS를 할당한다. 노드  $U_i^r$ 가 갖는 가중치 값이 크

**알고리즘 2. 가중치 우선순위에 따른 GTS 할당**

```

1 //Coordinator receives a  $GTS.request$  to  $U_i (i=1...N)$ 
2 for each node  $U_i$ 
3      $U_{set} = U_{set} \cup \{U_i\};$ 
4      $S = \emptyset;$ 
5     if  $count(GTS_{slot}) \geq count(U_{set})$ 
6          $S = S \cup \{U_i\};$ 
7     else
8         loop( $U_{set}$ )
9             if  $([RC_i^r - RA_i^r] > [RC_{i+1}^r - RA_{i+1}^r])$ 
10                 $S = S \cup \{U_i\};$ 
11            else
12                order_reverse( $U_i, U_{i+1}$ );
13                 $S = S \cup U_{set};$ 
14      $GTS.allocated(S);$ 

```

그림 7. 가중치 우선순위에 따른 GTS 할당 알고리즘  
Fig. 7. The GTS allocation algorithm based on weight priority.

면 다음 비컨 주기에서 GTS를 할당받을 확률이 높아지고, 반대로 값이 작으면 GTS를 할당받을 확률이 낮아진다. (그림 7)는 가중치 우선순위에 따른 GTS 할당 알고리즘을 나타낸다.

### V. 성능평가

#### 1. 실험 환경

(그림 8)은 제안하는 기법의 모의실험을 위해 구성된 [5-6] OPNET 시뮬레이터에서의 패킷 발생 모델을 나타낸다. 100m\*100m 네트워크 영역에 한 개의 PAN 코디네이터를 중심으로 7개의 센서 노드를 랜덤으로 배치한 성형 토폴로지로 구성하였다. 센서 노드는 CSMA/CA 과정에서 GTS 할당 요청에 대한 경쟁 노드(트래픽 유발 노드) 역할을 하는 1개의 *node\_CAP*와 단위시간 동안 GTS 할당 요청 및 데이터 전송을 위한 6개의 GTS 할당 요청 노드(*node\_GTS*)로 구성하였다. *node\_CAP*와 *node\_GTS*가 전송하는 패킷은 각각  $\lambda = 250$ 과  $\lambda = 300$ 을 발생 확률로 하는 Poisson 분포를 갖고 각 패킷은 고정된 길이를 갖는다고 가정한다. 특정 노드가 프레임을 전송하고 있으면 다른 노드들이 채널을 감지할 수 있도록 sensing sensitivity를 0으로 설정해, 허든 노드 문제를 해결하였다.

센서 노드 모델은 총 4가지 블록 모델로 구성하였다. (1)*Physical layer*은 2.4 GHz 대역폭을 갖는 무선 트랜시버(수신을 위한 *rx*와 전송을 위한 *tx*)로 구성하였다. (2)*MAC layer*는 PAN 코디네이터에 의해 주기적인 비

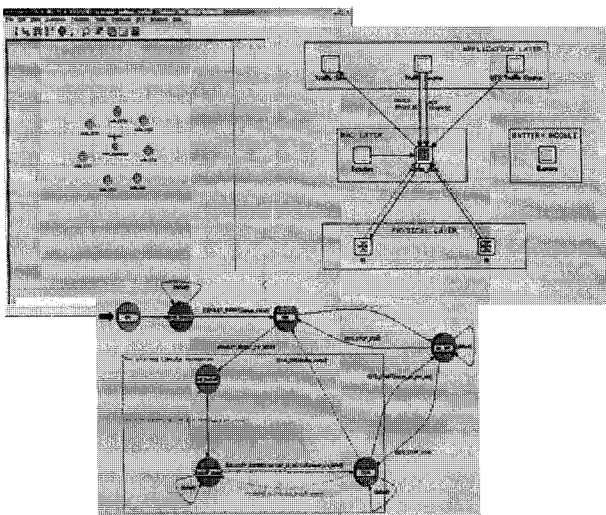


그림 8. 실험환경  
Fig. 8. The simulation test-bed.

표 1. 시스템 관련 파라미터  
Table 1. The value of system parameter.

파라미터	값
<i>aMaxBE</i>	5
<i>macMinBE</i>	3
<i>macMaxCSMABackoffs</i>	4
<i>aMaxFrameRetries</i>	3
<i>BO,SO</i>	2
<i>GTS slots length</i>	2
<i>Packet Size(MHR+MSDU)</i>	404 (104 + 300) bits
<i>ACK Frame</i>	11 bytes
<i>node_GTS</i> 패킷( $\lambda$ )	poission(300)
<i>node_CAP</i> 패킷( $\lambda$ )	poission(250)
<i>Duraion Time</i>	10 sec

컨을 발송하고 노드는 경쟁 기반인 slotted CSMA/CA를 사용해 통신하도록 하였다. (3)*Battery module*는 MICAz 모트 사양에 따른 잔여 에너지 및 소비 에너지 양을 측정하였다. (4)*Application layer*는 응답 프레임 생성을 위한 *Sensory data* 모듈과 *MAC command* 모듈로 구성하였다.

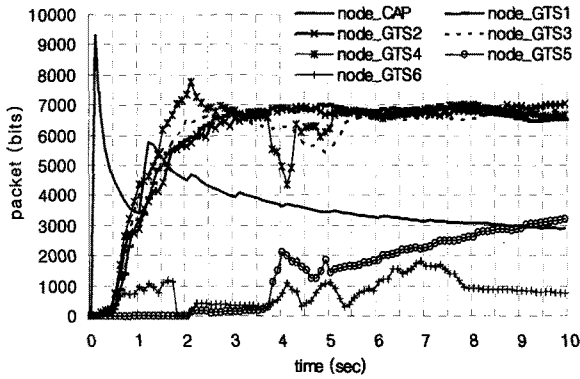
(표 1)은 실험을 위한 노드 설정 정보 및 주요 시스템 관련 파라미터 값을 나타낸다.

#### 2. GTS 요청 처리율

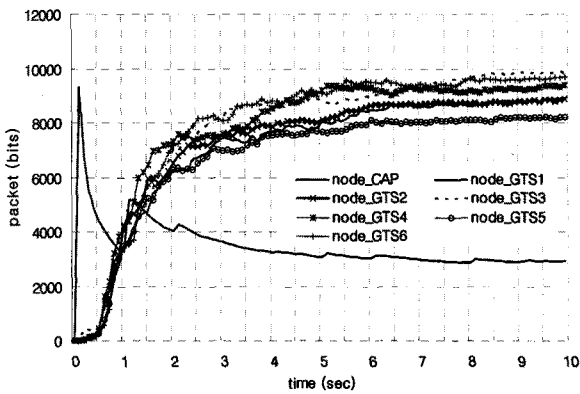
제안하는 GTS 할당 기법에 대한 GTS 요청 처리율(throughput)을 분석하기 위해 단위 시간당 전송 가능한 트래픽의 양을 측정하였다. (그림 9)과 (표 2)는 단위시간 동안 발생된 각 노드별 평균 패킷 전송량을 측정된 결과를 나타낸다.

실험 환경에서 각 노드들은 PAN 코디네이터에게 GTS 할당 요청을 하고, 다음 비컨 주기에서 2개의 GTS를 할당 받아 데이터를 전송한다. 하나의 비컨 주기에서 최대 가능한 GTS 할당 개수가 7개로 한정되어 GTS를 할당받지 못할 경우, 요청 재전송으로 인한 추가적인 오버헤드와 지연이 발생해 처리율이 감소된다.

측정 결과 표준 방식에서 *node\_GTS5*와 *node\_GTS6*의 경우, GTS를 할당받지 못해 단위시간 동안의 평균 패킷 전송량이 다른 노드들에 비해 약 35% 적게 나타났다. 또한 *node\_CAP*의 경우, 제안하는 방식과 동일한 패킷을 발생했음에도 불구하고 *node\_GTS5*와 *node\_GTS6*에 의한 GTS 할당 재요청으로 인해 약 13% 더 많은 패킷 전송량을 보이고 있다. 제안하는 방식의 경우, 각 노드의 평균 패킷 전송량은 일정함을 보인다. 즉, 모든 노드가 가중치를 이용해



(가) IEEE 802.15.4 평균 패킷 전송량



(나) 제안하는 가중치 기반 평균 패킷 전송량

그림 9. 노드별 평균 패킷 전송량

Fig. 9. The average throughputs of GTS.

표 2. 각 노드별 평균 패킷 전송량(bits/sec)

Table 2. The average throughputs of GTS per node.

구분	802.15.4	제안	증감구분
node_CAP	3,814.8	2,928.6	감소 ( 886.2)
node_GTS1	6,515.8	8,878.5	증가 (2,362.7)
node_GTS2	6,680.1	8,886.8	증가 (2,206.7)
node_GTS3	6,825.6	9,821.3	증가 (2,995.7)
node_GTS4	7,048.9	9,390.5	증가 (2,341.6)
node_GTS5	1,914.8	8,195.6	증가 (6,280.8)
node_GTS6	1,214.8	9,708.0	증가 (8,493.2)

공정하게 GTS를 할당 받음으로서 재전송으로 인한 처리 지연 및 오버헤드를 줄이고, 전송패킷 수가 증가하여 처리율이 약 60% 향상되는 결과를 보인다.

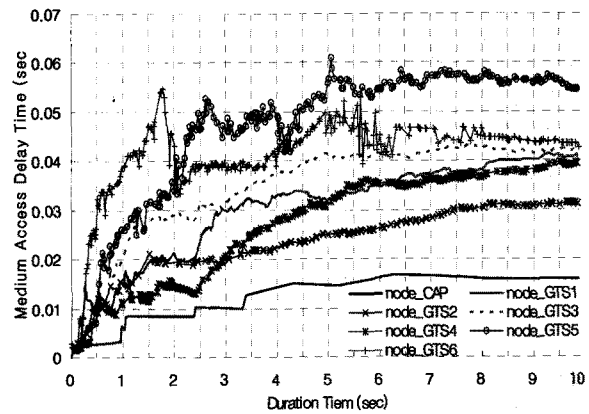
### 3. GTS 할당 지연 시간

제안하는 기법에 대한 GTS 할당 지연 시간(Delay time) 분석을 위해 GTS 할당 요청 후 실제 GTS를 할당받기까지의 채널 접근 지연 시간을 측정하였다. <sup>14</sup>

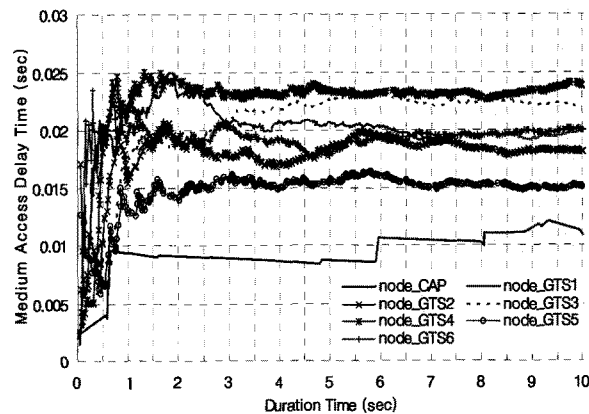
<sup>14</sup>GTS 할당 요청은 CFP 구간에서 이루어짐으로 CAMS/CA를 통한 채널 접근 과정은 GTS 성능과 밀접한 관련이 있다. CSMA/CA를 통해 채널에 접근하기 위해서는 임의로 정해진  $\min(2^{BE} - 1, macMaxBE)$  만큼의 backoff 지연과 2번의 CCA 수행 후 채널이 유희상태여야 한다. 실험 환경에서 최대 backoff 시도 횟수는 4회 이므로 채널을 할당받는데 필요한 총 backoff 심볼수는 다음과 같다.

$$L_{backoff} = 2 \left( \sum_{i=1}^4 (2_i^{BE} - 1) \right) + 2CCA \quad (15)$$

실험 환경에서  $BO, SO$  값이 2 이므로, 하나의 슬롯 당 심볼수는 240개이고, 심볼 당 비트수는 4이므로 슬롯 당 최대 960bits 데이터를 전송할 수 있다. 만일



(가) IEEE 802.15.4 채널 접근 지연 시간



(나) 제안하는 평균 채널 접근 지연 시간

그림 10. 노드별 GTS 할당을 위한 평균 채널 접근 지연 시간

Fig. 10. The average medium access delay time for GTS allocation.



표 3. 각 노드별 평균 채널 접근 지연 시간

Table 3. The average medium access delay time per node.

구분	802.15.4	제안	증감구분
<i>node_CAP</i>	0.016	0.009	감소 (0.007)
<i>node_GTS1</i>	0.034	0.023	감소 (0.011)
<i>node_GTS2</i>	0.025	0.018	감소 (0.007)
<i>node_GTS3</i>	0.036	0.022	감소 (0.014)
<i>node_GTS4</i>	0.029	0.024	감소 (0.005)
<i>node_GTS5</i>	0.052	0.014	감소 (0.038)
<i>node_GTS6</i>	0.049	0.019	감소 (0.030)

GTS 할당 요청 후 다음 비컨 주기에서 GTS를 할당받지 못할 경우, 최소한 현재 CFP 구간 크기(960\*7)와 다음 BI 구간(960\*16) 크기만큼 지연 시간이 발생한다. 즉, SO값이 증가할수록 단위 시간 당 처리할 수 있는 패킷 수는 많아지지만, 이에 따른 지연시간이 증가된다. (그림 10)와 (표 3)은 단위시간 동안 발생된 각 노드별 GTS 할당 요청에 대한 채널 접근 지연시간을 측정할 결과를 나타낸다.

(그림 9)의 GTS 요청 처리율 성능 분석에 따르면 *node\_GTS5*와 *node\_GTS6*의 평균 패킷 전송량이 다른 노드들에 비해 약 35% 적게 나타났다. 이와 관련해 두 노드의 평균 지연을 분석해보면 단위시간 동안의 평균 채널 접근 지연 시간이 각각 0.052, 0.049초로 다른 노드들에 비해 약 2배 더 많은 지연을 보이고 있다. 이러한 이유는 CSMA/CA에 의한 GTS 요청 할당 시 backoff로 인한 처리 지연이 발생하고, 다른 노드들에 비해 단위 시간동안 적은 횟수의 GTS 할당을 받음으로서 처리율 감소를 가져온다. 제안하는 방식의 경우, 전체 노드의 평균 채널 접근 지연 시간은 0.018초로 각 노드들의 채널 접근 지연 시간과 비교해 볼 때, 큰 오차를 보이지 않는다. 가중치 기반 공정 채널 할당을 통해 모든 노드들이 공정하게 채널을 할당 받음으로서 채널 접근 요청에 대한 지연 시간을 줄이고, 더 많은 패킷 전송을 함으로써 전체 서비스 처리 시간의 약 2배 향상시킨 결과를 가져온다.

## V. 결 론

IEEE 802.15.4 표준에 따른 GTS 할당 기법은 FIFO 스케줄링을 사용해 패킷을 할당함으로써 대역의 낭비가 심하고 채널 사용률을 감소시킨다. 본 논문에서는 공정 큐잉을 위한 가중치 기반 GTS 할당 기법을 제안하였다. 제안하는 기법은 각 노드의 GTS 할당 요청 횟수

(*RC*)와 최근 할당 여부(*RA*)를 이용한 가중치 기반 GTS 할당 기법을 통해 불공정성 문제를 해결함으로써 GTS 요청 처리율을 높이고 지연 시간을 감소시켰다. 시뮬레이션을 통해 제안하는 기법이 표준과 비교해 데이터 처리율과 채널 사용률 측면에서 효율적임을 보였다.

향 후 다양한 토폴로지 환경에서 노드 수 증가에 따른 보다 구체적인 성능 분석이 필요하다. 또한 실 장비를 대상으로 한 개발 및 구현을 통해 제안하는 기법의 보다 현실적인 검증이 필요하다.

## 참 고 문 헌

- [1] IEEE Std 802.15.4: "Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)", 2003.
- [2] Y. Huang, A. Pang, and H. Hung, "An adaptive GTS allocation scheme for IEEE 802.15.4". IEEE Transactions Parallel Distrib. Syst., pp. 641 - 651, 2008.
- [3] 임정섭(Jeong-Seob Lim), 윤완오(Wan-Oh Yoon), 서장원(Jang-Won Seo), 최한림(Han-Lim Choi), 최상방(Sang-Bang Choi), "LR-WPAN에서 충돌 회피를 위한 동적 채널할당 알고리즘", 대한전자공학회, 전자공학회논문지-TC, 제47권 TC편 제6호 2010.06, pp.10~21.
- [4] 김형석(Hyung Seok Kim), "IEEE 802.15.4 무선센서네트워크의 실시간 트래픽 처리를 위한 프레임/슬롯 할당방법," 대한전자공학회, 전자공학회논문지-CI, 제44권 제3호, 2007.5, pp. 49~56.
- [5] P. Jurcik, A. Koubaa, M. Alves, E. Tovar, Z. Hanzalek, "A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism," MASCOTS'07, October 2007, pp. 109-116.
- [6] Anis Koubaa, Mario Alves, Eduardo Tovar, "GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks," 14th WPDRTS, special track on Wireless Sensor Networks, Greece, 2006.
- [7] A. Koubaa, M. Alves, and E. Tovar, "i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4," Euromicro Conference on Real-Time Systems (ECRTS'06), Jul. 2006.
- [8] 이강우, 신연순, 안종석, "두 개의 우선순위 큐를 적용한 IEEE 802.15.4 GTS 서비스 지연에 대한 분석적 모델," 정보처리학회논문지C, vol.15C, no.5, 2008. 10, pp.419-428.

## 저 자 소 개



이 경 화(정회원)

2001년 가천의과대학 뉴미디어  
학과 학사 졸업.

2007년 숭실대학교 정보처리학과  
석사 졸업.

2007년~현재 숭실대학교 컴퓨터  
학과 박사과정 재학.

<주관심분야 : Wireless Sensor Network, Mobile  
Cimputing, Wireless LAN, QoS, Security>



이 협 건(학생회원)

2009년 3월~현재 숭실대학교  
컴퓨터학과 석사과정.

<주관심분야 : Wireless Sensor  
Network, Smart Grid, QoS>



신 용 태(정회원)

1985년 숭실대학교 한양대학교  
학사 졸업.

1990년 University of Iowa  
Computer Science  
석사 졸업

1994년 University of Iowa  
Computer Science  
박사 졸업

1994년 5월~8월 University of Iowa 객원교수

1994년 8월~1995년 1월 Michigan state  
University 객원교수

1995년 3월~현재 숭실대학교 교수

<주관심분야 : DRM, Multicast, QoS, Mobile  
Cimputing>