

## HR-WPAN을 위한 Worst-case Guaranteed Scheduling algorithm

Worst-case Guaranteed Scheduling algorithm for HR-WPAN

김 제민 이 정규  
( Je Min Kim Jong Kyu Lee )

**Abstract** – The proposed LDS(Link-status Dependent Scheduling) algorithm in HR-WPAN up to now aims at doing only throughput elevation of the whole network, when the crucial DEV(Device) is connected with worst-link relatively, throughput of this DEV becomes aggravation. The proposed the WGS(Worst-case Guaranteed Scheduling) algorithm in this paper guarantees throughput of the DEV which is connected with worst-link in a certain degree as maintaining throughput of all DEVs identically even if a link-status changes, decreases delay of the whole network more than current LDS algorithm. Therefore proposed WGS algorithm in this paper will be useful in case of guaranteeing throughput of a DEV which is connected worst-link in a certain degree in a design of HR-WPAN hereafter.

**Key Words** : WPAN, link-status, MAC scheduling, channel time scheduling, guaranteed scheduling

1. 서 론

현재까지 network의 throughput 향상을 위해 DEV의 link 상태를 이용한 LDS algorithm이 제안되었다. 그러나 좋은 link에 연결된 DEV는 많은 수의 TU(Time Unit)를 할당 받게 되어 throughput이 향상되지만, 나쁜 link에 연결된 DEV는 상대적으로 적은 수의 TU를 할당 받게 되어 throughput은 더욱 나빠지게 된다. 따라서 나쁜 link에 연결된 DEV가 상대적으로 중요한 DEV일 경우, 그 DEV의 throughput은 더욱 악화된다. 본 논문에서는 link 상태가 변하더라도 모든 DEV의 throughput을 동일하게 유지함으로서, worst-link에 연결된 DEV의 throughput을 보장하고 전체 network의 delay는 기존의 LDS algorithm보다 감소시킬 수 있는 WGS algorithm을 제안하였다.

## 2. 기존 LDS algorithm

현재까지 전체 network의 throughput 향상을 위해 제안된 LDS algorithm은 식(1)과 같이 나쁜 link에 연결된 DEV보다 좋은 link에 연결된 DEV에게 상대적으로 높은 가중치를 부여하여 많은 수의 TU를 할당하는 scheduling scheme을 사용하였다[1].

link 상태의 계산방법은 WGS algorithm의 식(3)~(5)의 방법과 동일하다. 각 DEV의 link 상태에 따른 가중치의 계산

방법은 식(1)과 같고 PNC(PicoNet Coordinator)가 DEV에게 할당 할 TU 개수는 식(2)의 방법으로 계산된다.

$$Q(i)_{n+1} = LS(i)_{n,r} / \sum LS(i)_{n,r}, \sum Q(i)_{n+1} = 1 \quad (1)$$

$$CTA(i)_{n+1} = \lfloor Q(i)_{n+1} \cdot CTAP/TU \rfloor \quad (2)$$

$Q(i)_{n+1}$	PNC에 의해 계산된 $n+1$ 번째 CTAP(Channel Time Allocation Period)에서의 $i$ 번째 DEV의 기종치
$CTA(i)_{n+1}$	$n+1$ 번째 CTAP에서 $i$ 번째 DEV에게 할당될 TU의 개수

### 3. WGS algorithm

### 3.1. link 상태 측정

DEV는  $n$ 번째 CTAP에서 전송한 frame에 대해 ACK frame의 수신 여부에 따라 식(3)과 같이 frame 전송에 성공할 경우에는 1의 값을 할당하고, 전송에 실패 할 경우에는 0의 값을 할당한다. 식(4)와 같이  $i$ 번째 DEV가 전송한 frame 수와 성공적으로 전송한 frame 수의 비율을  $i$ 번째 DEV의  $n$ 번째 CTAP에서의 link 상태라고 한다[2]. 만일  $n$ 번째 CTAP에서 DEV의 급격한 link 상태의 변화를  $n+1$ 번째 CTAP에 그대로 반영 할 경우, 잘못된 할당을 초래할 수 있다. 따라서 식(5)와 같이 DEV는 MA(Moving Average)를 이용하여  $i$ 번째 DEV의  $n-r+1$ 번째 CTAP부터  $n$ 번째 CTAP까지의 평균 link 상태를 계산한다[3]. 식(5)로부터 계산된 각 DEV의 평균 link 상태를 PNC에게 넘겨주고 PNC는 각 DEV의 link 상태에 따라 CTA를 할당 한다.

$$I(i)_n = \begin{cases} 1 & \text{success} \\ 0 & \text{failure} \end{cases} \quad (3)$$

$$LS(i)_n = frame(i)_n / CTA(i)_n \quad (4)$$

$$LS(i)_{n,r} = \begin{cases} \sum_{m=1}^{m=r} LS(i)_{n-m+1}/r, & n \geq r \\ \sum_{m=1}^n LS(i)_{n-m+1}/r, & n < r \end{cases} \quad (5)$$

$I(i)_n$	$i$ 번째 DEV가 $n$ 번째 CTAP에서 전송한 frame에 대한 전송 성공 여부
$frame(i)_n$	$i$ 번째 DEV가 $n$ 번째 CTAP에서 성공적으로 전송한 frame의 개수
$CTA(i)_n$	$n$ 번째 CTAP에서 $i$ 번째 DEV에게 할당된 TU의 개수
$LS(i)_n$	$n$ 번째 CTAP에서 측정된 $i$ 번째 DEV의 link 상태, $0 \leq LS(i)_n \leq 1$
$r$	MA(Moving Average) Range
$LS(i)_{n,r}$	$i$ 번째 DEV의 $n-r+1$ 번째 CTAP부터 $n$ 번째 CTAP까지의 평균 link 상태, $0 \leq LS(i)_{n,r} \leq 1$

### 3.2. worst-case guarantee를 위한 scheduling scheme

논문에서 제시한 WGS algorithm은 좋은 link에 연결된 DEV에게는 적은 수의 TU를 할당하고, 나쁜 link에 연결된 DEV에게는 상대적으로 많은 수의 TU를 할당하여 link 상태가 변하더라도 모든 DEV의 throughput이 동일하게 유지되도록 하는 scheduling scheme을 사용하였다. PNC가 각 DEV에게 할당해야 할 TU의 개수를 계산하는 방법은 예를 들어 다음과 같이 설명하였다. 3개의 DEV A, B, C가 있고, 각 DEV의 link 상태는  $LS(A)$ ,  $LS(B)$ ,  $LS(C)$ 이다. 그리고 PNC가 CTAP에서 각 DEV에게 할당하는 TU의 개수는  $CTA(A)$ ,  $CTA(B)$ ,  $CTA(C)$ 이다. 따라서  $n+1$ 번째 CTAP에서 PNC가 DEV에게 할당해야 할 TU의 개수는  $CTA(i)_{n+1}$ 이며,  $n+1$ 번째 CTAP의 link 상태는  $LS(i)_{n+1}$ 이다.

$DEV$	$A$	$B$	$C$
$LS(i)_n$	$LS(A)$	$LS(B)$	$LS(C)$
$LS(i)_{n+1}$	$LS(A)_{n+1}$	$LS(B)_{n+1}$	$LS(C)_{n+1}$
$CTA(i)_{n+1}$	$CTA(A)_{n+1}$	$CTA(B)_{n+1}$	$CTA(C)_{n+1}$

표 1. TU 개수 할당을 위한 예제

$n+1$ 번째 CTAP에서 DEV의 throughput을 성공적으로 전송한 frame의 수라고 할 때, DEV의  $n+1$ 번째 CTAP에서의 throughput은  $frame(i)_{n+1}$ 이므로  $n+1$ 번째 CTAP에서 각 DEV의 throughput이 같아지려면  $frame(i)_{n+1}$ 이 같아야 한다. 즉, PNC는  $n+1$ 번째 CTAP에서  $frame(A)_{n+1} = frame(B)_{n+1} = frame(C)_{n+1}$ 이 되는  $CTA(i)_{n+1}$ 를 각 DEV에게 할당해야 한다. 그리고  $frame(i)_{n+1} / CTA(i)_{n+1} = LS(i)_{n+1}$ 이므로 식(6)과 같이 쓸 수 있다.

$$CTA(A)_{n+1} \cdot LS(A)_{n+1} \quad (6)$$

$$= CTA(B)_{n+1} \cdot LS(B)_{n+1} = CTA(C)_{n+1} \cdot LS(C)_{n+1}$$

PNC가  $n+1$ 번째 CTAP에서 각 DEV에게 할당 할 TU의 개수,  $CTA(i)_{n+1}$ 의 합은 할당 가능한 TU의 개수와 같다고 가정한다.

$$CTA(A)_{n+1} + CTA(B)_{n+1} + CTA(C)_{n+1} = CTAP / TU \quad (7)$$

따라서 식(6), (7)로부터 DEV A에게 할당 해야 할 TU의 개수를 계산할 수 있다. 그러나 PNC는  $n+1$ 번째 CTAP에서 DEV의 link 상태를 미리 알 수 없으므로 다음과 같이  $LS(i)_{n,r} = LS(i)_{n+1}$  라고 가정하여 식(8)을 유도하였다.

$$CTA(A)_{n+1} = \left\lceil \frac{CTAP}{TU} / \left( LS(A)_{n,r} \cdot \sum \frac{1}{LS(i)_{n,r}} \right) \right\rceil \quad (8)$$

$LS(i)_{n+1}$	$n+1$ 번째 CTAP에서 $i$ 번째 DEV의 link 상태, $0 \leq LS(i)_{n+1} \leq 1$
$CTA(i)_{n+1}$	$n+1$ 번째 CTAP에서 $i$ 번째 DEV에게 할당될 TU의 개수

## 4. 성능 평가

### 4.1. 기본 모델

기본 모델은 HR-WPAN 표준에서 정의하고 있는 내용을 기반으로 설정되었으며, 성능 평가를 수행하기 위하여 다음의 가정들을 두고 실행 하였다. 첫째, 단일 Piconet만으로 구성하였다. 둘째, beacon 구간과 CAP는 고려하지 않았다[4].

Attribute	Value
total links	20
CTAP	65 ms
ACK policy	Imm-ACK
payload	256 Kbyte
data rate	55 Mbps
traffic type	FTP
MA(Moving Average) range	40
frame error prob. of best-link	0.02
simulation time	60 sec

표 2. 성능 평가 파라미터

### 4.2. 성능 평가

그림 1,2에서 LDS algorithm에서는 worst-link에 연결된 DEV가 PNC로부터 할당 받는 TU의 수가 감소하여 throughput이 매우 낮아졌으며 WGS algorithm에서는 worst-link에 연결된 DEV가 PNC로부터 할당 받는 TU의 수가 증가하고 best-link에 연결된 DEV가 PNC로부터 할당 받은 TU의 수가 감소하기 때문에 평균 throughput은 LDS algorithm보다 조금 낮게 나타났으나 worst-link에 연결된 DEV의 throughput은 LDS algorithm 보다 크게 향상 되었으며 DEV간의 throughput이 모두 동일함을 알 수 있다.

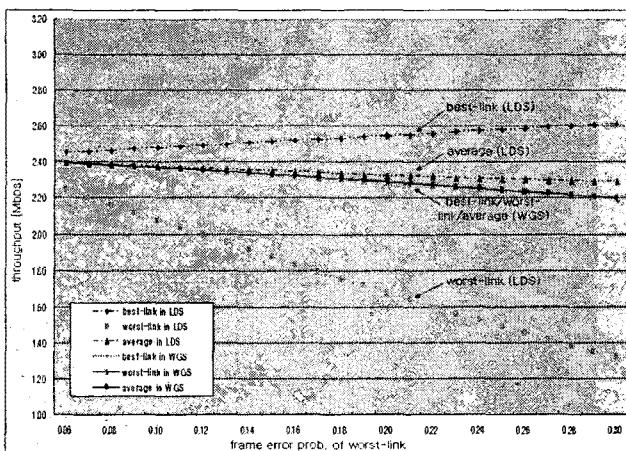


그림 1. worst-link의 frame error prob. 증가에 따른 throughput  
(best-link 15, worst-link 5, arrival rate 0.5 Mbyte/s)

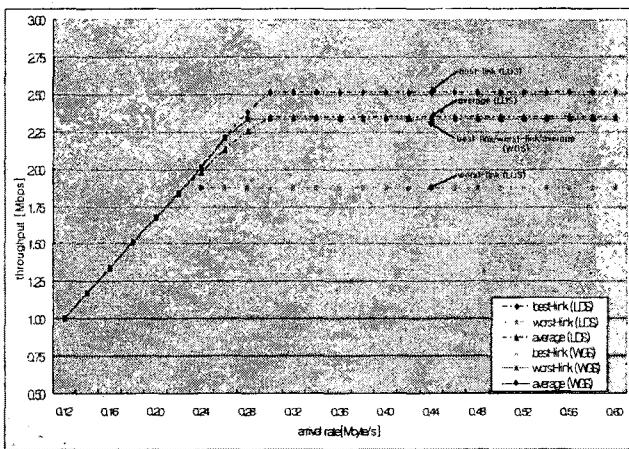


그림 2. arrival rate 증가에 따른 throughput  
(best-link 15, worst-link 5, frame error prob. worst-link 0.15)

그림 1,2에서 WGS algorithm의 평균 throughput은 LDS algorithm의 평균 throughput보다 조금 낮게 나타났으나 그림 3,4에서 평균 delay는 LDS algorithm보다 향상 되었으며 best-link에 연결된 DEV와 worst-link에 연결된 DEV의 delay가 모두 동일함을 알 수 있다.

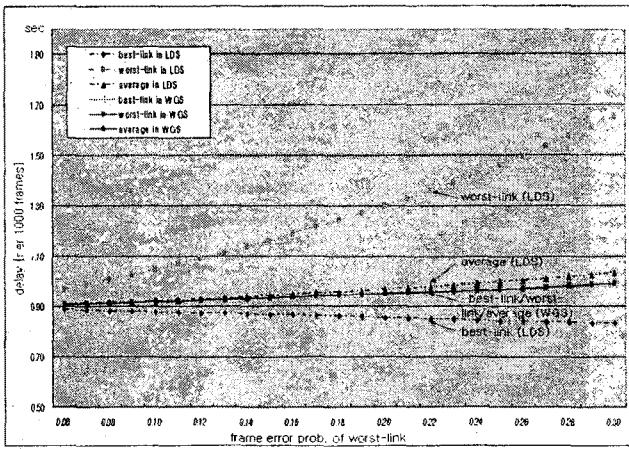


그림 3. worst-link의 frame error prob. 증가에 따른 delay  
(best-link 15, worst-link 5, arrival rate 0.5 Mbyte/s)

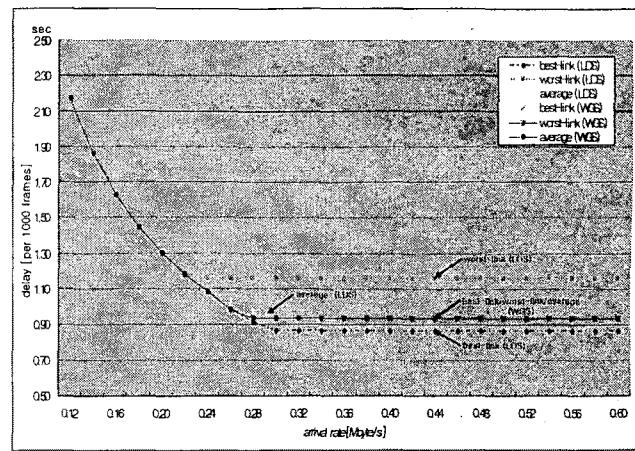


그림 4. arrival rate 증가에 따른 delay  
(best-link 15, worst-link 5, frame error prob. worst-link 0.15)

## 5. 결 론

성능평가 결과 제안한 WGS algorithm은 DEV간의 성능이 모두 동일하였으며 전체 network의 delay는 LDS algorithm 보다 향상되었다. 따라서 향후 coordinator나 base station에 의해 frequency/time division 방식으로 자원을 할당하는 IEEE 802.15.1 Bluetooth, IEEE 802.15.3 HR-WPAN, IEEE 802.15.4 LR-WPAN, IEEE 802.16 WiMAX의 설계시 worst-link에 연결된 DEV의 throughput을 어느 정도 보장하고 전체 network의 delay를 감소시키고 싶을 경우, WGS algorithm이 매우 유용하게 사용될 수 있을 것으로 기대된다.

## 참고 문헌

- [1] Dong-won Kwak, Seung-hyong Rhee, "Location Dependent Scheduling in 802.15.3 High-rate WPAN", the KICS vol.29, 2004
- [2] Byung-Seo Kim, Yuguang Fang and Tan F.Wong, "Rate-Adaptive MAC Protocol in High-Rate Personal Area networks", IEEE WCNC, Mar. 2004
- [3] Robert A. Malaney, Ernesto Exposito, Xun Wei and Dao Trong Nghia, "Predicting Location-Dependent QoS for Wireless Networks", LNCS Vol. 3479, pp. 328 - 340, 2005.
- [4] WoongChul Choi, KwangSue Chung, Seung Hyoung Rhee, and Jin-Woong Cho, "WFCTA (Weighted Fair Channel Time Allocation) and Its Analysis for HR-WPAN", LNCS Vol. 3563, pp. 358 - 367, 2005.