

# 블루투스의 효율적 스캐터넷 동작을 지원하기 위한 MAC 스케줄링 기법

정희원 주 양 익\*, 김 용 석\*, 이 태 진\*\*, 차 균 현\*

## MAC Scheduling Algorithm for Supporting Efficient Scatternet Operations of Bluetooth

Yang-Ick Joo\*, Yongsuk Kim\*, Tae-Jin Lee\*\*, and  
Kyun Hyon Tchah\* *Regular members*

요 약

본 논문에서는 블루투스(Bluetooth) 시스템의 스캐터넷(scatternet) 동작을 지원하기 위한 효율적인 MAC (Medium Access Control) 스케줄링(scheduling) 알고리즘을 제안한다. 특히 기존에 제안된 스캐터 모드(scatter mode)의 문제점을 개선하여 각 링크별 QoS (Quality of Service) 요구조건을 고려한 방안을 제시한다. 스캐터 모드는 기본적으로 스캐터넷 상의 연결된 모든 링크에 대해 동일한 서비스 기회를 제공하기 위한 방안으로, 시분할 방식을 기반으로 한다. 하지만 각각의 블루투스 기기들은 링크마다 서로 다른 트래픽 특성을 가지며, 그 양 또한 서로 다르므로 모든 링크에 대해 동일한 서비스 기회를 제공하는 것은 효율적인 서비스 슬롯(slot)의 활용이 되지 못하며 많은 무선 자원의 낭비를 가져오게 된다. 따라서 본 논문에서는 표준에서 제시하고 있는 기존의 스캐터 모드 방식을 유지하며, 상대적으로 무선 자원의 사용 효율을 높이고 각 링크마다의 트래픽 특성 및 QoS 요구조건을 수용할 수 있는 새로운 알고리즘을 제안하며, 시뮬레이션 결과를 통해 그 성능을 검증한다.

### ABSTRACT

In this paper, we propose an efficient MAC scheduling algorithm for supporting Bluetooth scatternet operations. Our proposed algorithm improves overall performances of scatter mode [2, 3] and considers QoS requirements of each link of the bridge node of a scatternet. Scatter mode is based on time division scheduling and provides every links of a scatternet with fair service opportunities. However, since each Bluetooth devices has various traffic characteristics, fair allocation of service opportunities is not efficient and may cause extravagance of wireless resources. Therefore, we propose a new efficient inter-piconet scheduling algorithm compliant with the current Bluetooth specification [1], and demonstrate its improved performances via computer simulations.

### 1. 서 론

블루투스는 2.4 GHz ISM 대역(Industrial Scientific Medical band)을 이용하여 저전력, 저가의 특성을 갖는 좁은 영역에서의 무선 통신 기술로서, 각 전자 기기들간의 연결 케이블(cable)을 대체하기 위한 방안으로 제안된 기술

이다. 이와 같은 무선 개인 통신망 기술(Wireless Personal Area Network: WPAN)은 동종 및 이종 기기들간의 무선 채널을 통한 효율적 연결을 제공함으로써 다양한 분야에서 큰 관심을 모으고 있으며, 특히 노트북, PDA, 이동전화, 헤드셋(headset) 등 다양한 기기들간의 유기적인 연결을 제공할 것으로 기대되고 있다.

\*고려대학교 전자공학과 (yiju@korea.ac.kr)

\*\*성균관대학교 정보통신공학부

논문번호: 020492-1111, 접수일자: 2002년 11월 11일

블루투스는 TDD 방식을 기반으로 하며 마스터(Master)-슬레이브(Slave) 구조를 갖는다. 슬레이브는 반드시 마스터가 전송한 직후의 슬롯에서만 전송이 가능하며, 마스터는 짝수번째 슬롯, 슬레이브는 홀수번째 슬롯을 사용하도록 할당되므로 블루투스 시스템의 MAC 스케줄링은 항상 피코넷(piconet)에서의 마스터-슬레이브 쌍을 단위로 이루어진다. 한편 범위를 확장하여 하나 이상의 피코넷이 연결된 스캐터넷(scatternet) 측면에서의 블루투스 동작을 살펴보면, 여러 피코넷에 포함된 PMP (Participant in Multiple Piconets) 노드는 시분할 방식을 기반으로 각 피코넷에 통신하는 시간을 할당하게 된다. 이 경우 단지 라운드로빈(round-robin) 방식으로 각 피코넷마다 한번씩 폴링(polling)하게 된다면, 각 피코넷들이 서로 동기가 맞지 않으므로, 각 피코넷을 스위칭(switching)하는 동안 보호구간(guard time)에서의 시간 슬롯이 낭비될 것이며, 또한 전송할 데이터가 없는 경우에도 폴링되어서 서비스 슬롯의 낭비를 초래하게 된다.

일반적으로 블루투스는 기기의 큰 이동성으로 인해 그 응용분야가 애드혹(ad hoc) 네트워크의 형태를 포함할 것으로 예상되며, 따라서 다수의 블루투스 피코넷을 연결하여 스캐터넷을 형성하게 되는 형태를 예상할 수 있다. 하지만 현재 블루투스 표준 1.1 [1]에서는 스캐터넷에 대해 단지 간략한 소개 수준으로만 언급되어 있는 실정이다. 최근 들어 [2, 3, 5-7] 등에서 스캐터넷을 지원하기 위한 스케줄링 방안이 제안되었으며, 이 중 [2, 3]에서 제안된 스캐터 모드가 현재 표준으로 제시되고 있다. 하지만 스캐터 모드는 각 링크마다의 형평성(fairness) 측면에서의 성능을 향상시키기 위한 방안이므로, 서로 다른 트래픽 특성을 갖는 각 링크의 성능을 효율적으로 보장할 수 없다. 따라서 각 링크의 트래픽 특성과 QoS 요구사항을 반영하여 효과적으로 서비스 슬롯을 각 피코넷에 할당할 수 있는 스캐터넷 스케줄링 알고리즘이 요구되며, 본 논문에서는 이러한 사항을 고려한 효율적인 스캐터넷 스케줄링 알고리즘을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 제안된 블루투스 시스템의 스캐터넷 동작을 지원하기 위한 스캐터 모드와 이를 위한 스케줄링 방안인 credit 방식에 대해 검토해보고, 3장에서는 이에 대한 문제점을 극복하기 위해

제안한 알고리즘이 설명될 것이다. 4장에서는 컴퓨터 시뮬레이션을 통한 제안한 알고리즘에 대한 성능분석과 기존의 알고리즘과의 성능 비교가 수행되고, 끝으로 5장에서 결론을 맺는다.

## 2. 블루투스 스캐터넷

### 2.1 스캐터 모드

스캐터 모드 [2, 3]는 새롭게 정의된 블루투스의 연결모드로, 블루투스 기기가 시분할 방식의 기반으로 동시에 여러 피코넷에 참여하는 것이 가능하도록 하며, 이를 통해 블루투스의 스캐터넷 동작을 지원한다. 스니프 모드와 유사한 점이 있으나, 스캐터 모드는 근본적으로 전력 소모를 줄이기 위한 목적으로 정의되지는 않았다. 하지만 PMP 기기는 스캐터 모드를 통해 피코넷 간의 시분할 다중화를 함으로써 휴지(idle) 구간을 최소화하여 결국 전력소모를 줄이는 효과를 얻을 수 있다. 스캐터 모드에서 블루투스 기기는 서로 다른 피코넷에서 다른 역할을 수행할 수 있으며, 이 모드는 시분할 방식을 통해 피코넷 사이에서의 스위칭을 보다 쉽고 효율적으로 하기 위해 정의되었다.

스캐터 모드에서의 스캐터넷 스케줄링 알고리즘은 서로 동기가 맞지 않는 피코넷들 사이의 보호구간의 낭비와 각 슬롯에서의 해당 피코넷에 대한 존재여부(presence)의 결정에 대한 복잡도, 트래픽 양의 변화에 대한 적응성, 클럭 전이(clock drift) 등에 대한 문제점을 안고 있다. 이러한 문제점들을 해결하기 위해 스캐터넷 스케줄링에서는 엄격한 통신 스케줄을 정의하지 않고 통신이 시작될 수 있는 시점을 presence point로 정의하여 해당 피코넷과 통신하며 통신 기간은 현재 링크의 사용정도나 교환되는 데이터의 양에 의해 결정하도록 한다. 사전에 결정된 스케줄이 없으므로 각 기기는 마스터의 presence point의 타이밍을 관찰하여 클럭 전이를 조절하게 된다.

#### 2.1.1 Presence Points (PPs)

스캐터넷 스케줄링에서의 주된 관점은 PMP 노드에 연결된 기기들이 동시에 동일한 피코넷에

존재(presence)할 수 있도록 하는 것이다. 즉, 피코넷에서는 마스터가 폴링을 하게 되면 항상 슬레이브는 해당 피코넷에서 마스터의 폴링을 기다리고 있기 때문에 즉시 응답할 수 있지만, 스캐터넷에서는 해당 피코넷에 대한 존재 시점이 일치하여야 통신이 가능하므로 이 타이밍이 어긋나지 않도록 하는 것이 중요한 문제점이다. 따라서 이러한 존재 시점을 일치시키기 위한 장치가 바로 Presence Point (PP)이다.

PP는 마스터와 슬레이브가 해당 피코넷에서의 존재여부와 메시지 교환을 원하는지 여부를 결정하는 시점이다. 스캐터 모드에서의 각 링크의 두 기기는 서로 PP의 패턴을 알고 있으며, 이 PP에서의 해당 피코넷에 대한 존재여부는 강요되지 않는다. 따라서 PP에서의 응답을 받지 못한 것을 링크 상황이 좋지 않은 것으로 받아들일 필요는 없다. PP는 짝수번째 슬롯에서 시작되는 두 슬롯으로 구성되며, 스캐터 모드에서 마스터가 슬레이브와 통신하고 싶다면, PP의 첫번째 슬롯에서 마스터가 슬레이브에게 presence signaling message를 전달하면 된다. 만약 슬레이브가 presence signaling PDU를 수신한다면 바로 다음의 홀수번째 슬롯에서 이에 대해 응답을 한다. 이러한 presence signaling PDU의 교환 이후에, 두 기기들 사이에서 Communication Event (CE)가 시작되며, 이는 두 기기가 서로 정보를 교환하는 구간이다.

### 2.1.2 Communication Events (CEs)

PP에서 signaling PDU가 성공적으로 교환되면 PP 직후의 Master-to-Slave 슬롯에서 Communication Event (CE)가 시작된다. CE 구간 동안 두 기기는 동일 피코넷에서 동작하며 정해진 CE의 최대 길이를 초과한 경우, 두 기기 모두 더 이상의 전송할 데이터가 없거나 전송이 허용되지 않는 경우, 혹은 특정 슬롯(N<sub>ce\_timeout</sub>)동안 어떠한 데이터도 수신하지 못하는 경우의 상태까지 통신을 지속한다.

CE의 최대 길이(Maximum Length)는 presence signaling PDU가 교환되면서 각 기기는 상대방 기기와 통신을 지속하고자 하는 최대 슬롯 수를 전달하며, 이 과정을 통하여 자신의 값과 상대 기기의 최대값 중 최소값을 취하여 해당 CE의 최대 길이를 결정한다. CE는 각 기기들이

POLL-NULL 시퀀스(POLL 패킷 혹은 NULL 패킷 단독의 경우 제외)를 감지하면 즉시 중단되고, 마스터와 슬레이브의 presence signaling PDU의 E 플래그(flag)에 대한 해당 비트가 set된 경우, PP 이후 즉시 통신을 중단하게 되며, E 플래그는 전송할 데이터가 없는 경우를 표시한다. 또한 한 기기가 더 이상 새로운 데이터를 수신할 수 없는 경우에도 CE는 중단될 수 있으며, 기저대역(baseband) 패킷 헤더(header)의 FLOW 비트가 clear됨으로써 표시된다. 이 경우 상대방 기기는 POLL 혹은 NULL 패킷만을 전송할 수 있다.

### 2.2 credit 방식

PP는 스캐터 모드에서 마스터와 슬레이브 사이의 통신이 시작되는 시점을 의미한다. 하지만, 두 기기 모두 이 PP에서의 존재여부는 선택사항이므로 이를 효율적으로 관리하기 위한 스케줄링 알고리즘이 필요하며, [2, 3]에서는 스캐터 모드의 스케줄링을 위해 credit 방식을 제안하였다.

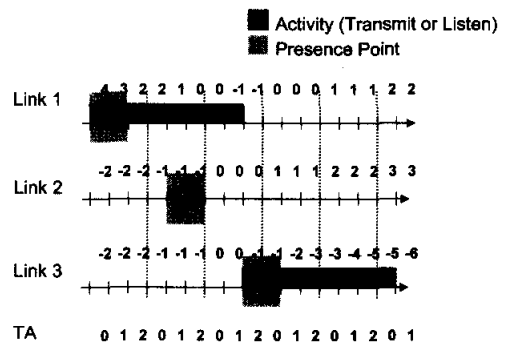


그림 1. credit 방식의 예 [2].

스캐터 모드에 있는 PMP 노드에서는 이전까지 불공평하게 서비스 기회를 제공받은 링크는 상대적으로 높은 우선순위를 부여받도록 하여 상대적인 서비스 기회에 대한 형평성을 제공하기 위해, 기기마다의 각 링크에 대해 각기 사용할 수 있는 슬롯의 수를 계산하여야 한다. 이를 credit으로 표현하며, 한 credit은 해당 링크에서 한 슬롯을 사용할 수 있음을 나타낸다. 그림 1을 살펴 보면, CE가 진행중인 링크에 대해서는 매 슬롯마다 credit이 1씩 감소하며, 전체 credit

의 합을 일정하게 유지하기 위해 TA (Temporary Account)를 두어서 매 슬롯마다 한 credit씩 증가시킨다. 알고리즘이 처음에 '0' credit으로 시작되므로 credit의 전체 합은 항상 '0'으로 일정하다. 해당 기기의 링크 수를  $n$ 으로 가정할 때, TA의 값이  $n$  credits이 되면, TA의 값은  $n$ 만큼 감소되고, 각 링크의 credit은 1씩 증가된다.

TA의 값을 제외한 나머지 credit은  $N_{max\_credit}$ 으로 그 값이 제한되며, 이를 통해 링크별 최대 버스트(burst)를 제한한다. 그리고, 새로운 링크가 추가되었을 때 새 링크의 credit은 '0'으로 초기화된다. 반대로 한 링크가 끊어진 경우, 그 링크의 credit은 TA에 추가되어서 나머지 링크에 동일하게 분배되게 된다.

진행중인 CE에 대해 새로운 PP에서의 credit이 보다 높은 값일 경우, 또는  $T_{scatter\_poll}$  슬롯동안 폴링되지 않은 경우에는 진행중인 CE가 중단되게 되고, POLL-NULL 시퀀스나 전송 오류로 인해 타임아웃되어 CE가 일찍 중단된 경우 다음의 PP를 미리 연결하여 credit과 관계없이 새로운 CE가 시작되도록 한다.

**2.3 최소 서비스 시간(Minimum Service Time)과 스위치 문턱값(Switch Threshold)**

지금까지 설명된 credit 방식은 피코넷 간의 잦은 스위칭을 야기할 수 있다. 잦은 스위칭은 매 스위칭마다의 보호구간을 고려하게 되면 많은 슬롯의 낭비가 발생할 수 있으므로 스위치 횟수를 최소화하여야 한다. 따라서 스캐터 모드에서는 잦은 스위칭을 방지하기 위한 두 가지 방안이 제안되었다.

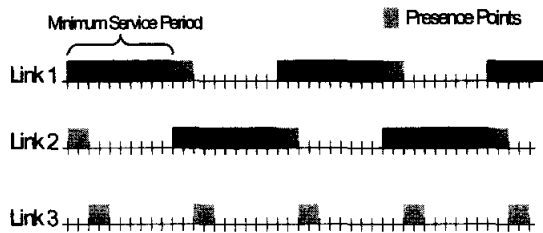


그림 2. 최소 서비스 시간의 starvation 문제 [2].

첫째 최소 서비스 시간(Minimum Service Time)을 보장하는 방안이 가능하다. 하지만, 이

는 한 링크의 PP가 다른 링크의 최소 서비스 시간동안 주기적으로 발생하는 경우, 그림 2와 같은 starvation 문제가 발생할 수 있다. 따라서 이러한 문제를 극복하고 잦은 스위칭을 방지하기 위해 새로운 링크에 대한 credit이 기존 링크의 credit 보다 특정 문턱값,  $N_{switch\_th}$ 만큼 커야 진행되는 CE가 중단될 수 있도록 하는 방안이 가능하다.

**2.4 credit 재분배(redistribution)**

POLL-NULL 시퀀스, 혹은 presence signaling PDU의 E 플래그가 set되어 CE가 중단된 경우, 이 링크의 credit은 다른 링크들에게 분배된다. 재분배되는 credit의 양은 식(1)에 의해 계산된다.

$$\max\left(\left[\frac{n-1}{n}(ac_{red} - ac_{min})\right], \left[\frac{n-1}{n}(ac_{red} - ac_{max} + N_{switch\_th})\right]\right) \quad (1)$$

여기서  $ac_{red}$ 는 재분배되는 링크의 credit이고,  $ac_{min}$ 은 최소 credit을 갖는 링크의 credit이며,  $ac_{max}$ 는 재분배되는 링크를 제외한 링크 가운데 최대 credit을 갖는 링크의 credit을 나타낸다.  $n$ 은 해당 기기에 연결된 링크의 수이다.

재분배되는 링크의 credit을 위에서 계산된 양만큼 줄이고, 이 양만큼의 credit을 나머지 링크에 균일하게 분배하여 재분배가 이루어진다. 물론 분배되는 credit의 양이 양수인 경우에만 재분배가 이루어진다.

**3. 적응적 credit 재분배(Adaptive Redistribution of Credits: ARC)**

2절에서는 [2, 3]에서 제안된 스캐터 모드에 의한 블루투스 스캐터넷 스케줄링 알고리즘, 즉 credit 방식에 대해 살펴 보았다. 이 방식은 블루투스 스캐터넷에서의 PMP 노드의 각 링크에 대해 동일한 서비스 기회를 제공할 수 있는 효과적인 방안이라 할 수 있지만, 각 마스터-슬레이브 쌍, 혹은 PMP 노드에 연결된 각 피코넷마다의 트래픽 특성에 의한 효율적인 스케줄링은 지원할 수 없는 단점이 있다. 특히 QoS 측면에서의 성능을 고려한다면 기존의 방식으로는 다양한 데이터 트래픽 특성을 고려하여 각 링크에 효과적으로 자원을 할당하기가 어렵다. 따라서 본 절에서는 PMP 노드의 각 피코넷마다의

QoS 요구사항을 고려한 효율적인 스캐터넷 스케줄링 알고리즘을 제안하고자 한다.

기존의 credit 방식에 있어서 PMP 노드의 각 링크마다 credit이 분배되는 경우는 두 가지가 존재한다. 하나는 TA의 credit이 해당 PMP 노드에 연결된 링크의 수와 같아지면 그 credit을 연결된 링크에 동시에 균일하게 분배하는 경우이고, 두번째는 POLL-NULL 시퀀스에 의해 CE가 중단되면 해당 링크의 credit이 나머지 링크들에게 동시에 균일하게 배분되는 경우이다. 두 경우 모두 서비스 기회의 형평성 측면에서 본다면 타당한 방식이라 할 수 있지만, PMP 노드에 연결된 각 링크, 즉 각 피코넷은 서로 소통되는 트래픽의 특성이 다르기 때문에 잉여 credit의 균등 분배는 효율적이라 볼 수 없다. 즉, 기존의 구조로는 각 링크별 QoS를 지원하기 어렵다.

따라서 본 절에서는 기존 스캐터 모드의 이러한 한계를 극복하고 각 링크별 QoS를 지원할 수 있는 적용적 credit 재분배(Adaptive Redistribution of Credit: ARC) 방식을 제안한다. ARC 알고리즘에서는 앞서 설명한 두 가지 경우에서의 잉여 credit 분배에 있어서 PMP 노드에 연결된 각 링크별 트래픽 특성, 혹은 요구되는 QoS 조건을 고려한다. 이를 위한 기준으로 스캐터넷 환경에서의 대표적인 QoS 파라미터라 할 수 있는  $T_{scatter\_poll}$ 을 사용하기로 한다. 스캐터넷이 형성되면서 피코넷간에 협의되는  $T_{scatter\_poll}$  파라미터에 의해 존재해야 하는 PP가 결정되므로, PMP 노드의 각 링크마다 최소한으로 필요한 PP를 이 파라미터를 이용한 스케줄링으로 지원할 수 있다. 즉,  $T_{scatter\_poll}$ 이 길수록 해당 링크에서 전송할 트래픽의 양이 상대적으로 적으며, 짧을수록 전송할 트래픽의 양이 상대적으로 많다는 것을 의미하므로, 제안한 ARC 방식에서는 PMP 노드에 연결된 각 링크마다의  $T_{scatter\_poll}$ 의 역수의 비로써 잉여 credit 중 해당 링크에 분배해야 할 credit의 양을 결정하도록 한다. 따라서 다음 절에서 잉여 credit 할당에 대한 두 가지 경우에 대해 그 세부적인 과정을 제안하고자 한다.

### 3.1 경우 I: 링크수와 TA의 credit이 같은 경우

경우 I에 대한 ARC 방식을 설명하기에 앞서  $mir(\cdot)$ 라는 새로운 연산자를 정의한다. 이 연산자는 피연산자들 사이의 최소 정수비를 구하는 연산자이다. 따라서 식(2)와 같이 정의될 수 있다.

$$mir(x_1, x_2, \varphi, x_n) = \frac{L}{x_1} : \frac{L}{x_2} : \varphi : \frac{L}{x_n} \quad (2)$$

여기서  $L=LCM(x_1, x_2, \dots, x_n)$ 은  $x_1, x_2, \dots, x_n$ 의 최소공배수(Least Common Multiple: LCM)를 나타내며, 각  $x_i$ 는 정수이다.

우선 각 링크들이  $T_{scatter\_poll}$ 에 대해 오름차순으로 정렬되며, 그 순서대로 1부터  $n$ 까지 링크마다 번호를 붙인다면, 가장 짧은  $T_{scatter\_poll}$ 이 첫 번째 링크로, 가장 긴  $T_{scatter\_poll}$ 이  $n$ 번째 링크로 대응된다. ARC 방식에서는 PMP 노드에서의 각 링크에 대한  $T_{scatter\_poll}$ 의 최소 정수비가 앞에서 정의된  $mir(\cdot)$  연산자에 의해 식(3)과 같이 계산된다.

$$mir(T_{scatter\_poll,1}, T_{scatter\_poll,2}, \varphi, T_{scatter\_poll,n}) = \frac{L}{T_{scatter\_poll,1}} : \frac{L}{T_{scatter\_poll,2}} : \varphi : \frac{L}{T_{scatter\_poll,n}} = a_1 : a_2 : \varphi : a_n \quad (3)$$

TA의 credit이  $t$ 가 되면, 즉 앞서 계산된 각 정수비의 합과 같아지면 ( $t = a_1 + a_2 + \varphi + a_n$ ), 이  $t$  credit은 각 링크마다  $T_{scatter\_poll}$ 의 역수비에 의해 적용적으로 분배된다. 즉, 식(4)와 같이 분배된다.

$$(a_1, a_2, \varphi, a_n) \rightarrow (link\ 1, link\ 2, \dots, link\ n) \quad (4)$$

만약 스캐터 모드의 credit 방식에서와 같이  $n$  credit으로  $T_{scatter\_poll}$ 의 역수비에 의해 분배한다면, 일반적으로 PMP 노드에 연결된 링크의 수,  $n$ 은 매우 작은 정수이므로 긴  $T_{scatter\_poll}$ 을 갖는 링크들은 '0' credit을 할당받을 수 있고, 이로 인해 공정하지 못하고, 비효율적인 슬롯 할당이 이루어질 수 있다. 따라서 본 논문에서 제안하는 ARC 스케줄러는 TA의 credit이 전체 정수비의 합  $t$ 와 같아진 후 credit을 분배하도록 한다.

### 3.2 경우 II: POLL-NULL 시퀀스에 의한 잉여 credit의 발생

POLL-NULL 시퀀스가 발생하거나, presence 시그널링에 의해 CE가 중단되는 경우 해당 링

크의 잉여 credit이 나머지 링크들에게 분배된다. 이는 사용되지 않은 credit을 보다 많은 대역을 요구하는 링크에게 재분배하는 의미를 가지며, 재분배되는 잉여 credit의 양은 식(1)에서와 같이 정의되며, 이를 식(5)와 같이  $c$  credit으로 나타내기로 한다.

$$c = \max \left( \left[ \frac{n-1}{n} (ac_{red} - ac_{min}) \right], \left[ \frac{n-1}{n} (ac_{red} - ac_{max} + N_{switch\_th}) \right] \right) \quad (5)$$

경우 I에서와 같은 방식으로 우선 각 링크들이  $T_{scatter\_poll}$ 에 대해 오름차순으로 정렬되며, 그 순서대로 1부터  $n$ 까지 링크마다 번호를 붙인다면, POLL-NULL 시퀀스가 발생한 즉시 식(5)에 의해 계산된  $c$  credit을 나머지 링크에 다음과 같이 적용적으로 배분한다.

POLL-NULL 시퀀스가 발생한 링크를  $j$ 번째 링크로 가정하고, 그 링크가 아닌  $k$ 번째 링크에 대한 가중치(weight),  $w_k$ 를 식(6)과 같이 정의한다.

$$w_k = \frac{T_{scatter\_poll,k}^{-1}}{\sum_{i=1, i \neq j}^n T_{scatter\_poll,i}^{-1}}, \quad k = 1, 2, \dots, n, \quad k \neq j \quad (6)$$

그리고 사용되지 않은 잉여  $c$  credit은 각 링크의 QoS 요구조건을 고려하여 각 링크별 가중치에 따라 식(7), (8)과 같이 배분된다.

$$\lfloor cw_k \rfloor \rightarrow \text{link } k, \quad k = 2, \dots, n, \quad k \neq j, \quad (7)$$

$$\left( c - \sum_{i=1, i \neq j}^n \lfloor cw_i \rfloor \right) \rightarrow \text{link } 1 \quad (8)$$

여기서  $\lfloor x \rfloor$ 는  $x$ 를 넘지 않는 최대 정수를 나타내며, 따라서 잉여 credit은 POLL-NULL 시퀀스가 발생한 링크와 첫번째 링크, 즉  $T_{scatter\_poll}$ 이 가장 짧은 링크를 제외한 각 링크마다의 가중치를 고려하여 적용적으로 배분되며, 그 나머지 credit이 첫번째 링크에 할당되게 된다.

#### 4. 시뮬레이션 결과

##### 4.1 ARC 방식

3절에서 제안한 알고리즘을 바탕으로 다음과 같은 환경에서의 시뮬레이션을 통해 ARC 방식의 성능을 검증한다. 우선 그림 3에 시뮬레이션을 위한 토폴로지(topology)를 도시하였다. 가운

데의 PMP 노드를 중심으로 세 개의 피코넷이 연결되어 있으며, 각 노드에서의 트래픽 발생은 소스(source) 노드와 목적지(destination) 간의 end-to-end 데이터 전송을 고려하였고, 그 흐름을 그림에 표시하였다. 전송되는 패킷은 1 슬롯 크기의 패킷만을 고려하였으며, 각 흐름에 대해서는 flow 0과 1은 전체 시뮬레이션 시간동안 데이터가 전송되도록 하였고, flow 2는 10000 슬롯부터 지속적으로 데이터가 전송되며, flow 3는 처음부터 24000 슬롯동안 데이터가 전송됨을 가정하였다. PMP 노드의 각 피코넷으로의 링크를 각각 link 0, link 1, link 2로 하였을 때, 각 링크의  $T_{scatter\_poll}$ 의 비는 link 0에서 link 2까지 차례로 1:2:3으로 가정하여, 각 링크에 대한 credit 분배를 QoS를 지원하기 위해 앞서 제안한 ARC 방식에 따라 링크마다 각각 6:3:2로 분배하였다.

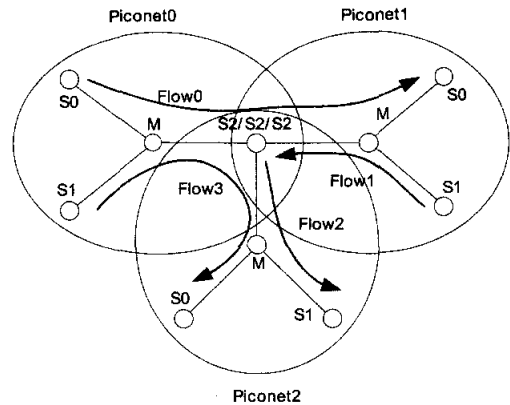


그림 3. 시뮬레이션 환경.

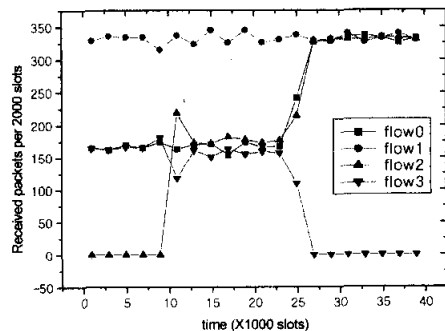


그림 4. 스캐터 모드의 서비스 시간에 따른 흐름별 수율 변화(ARC 방식 적용하지 않음).

그림 4와 그림 5에서는 ARC 방식을 적용한 경우의 성능을 평가하기 위하여 서비스 시간을 증가시키면서 일정 구간에서 수신되는 패킷의 수를 관찰하였다. 그림 4를 보면, ARC 방식을 적용하지 않은 일반적인 경우 비교적 링크상의 전송되는 패킷 수가 적은 flow 1이 가장 큰 수율을 나타내었고, 가장 트래픽 양이 많은 flow 0의 경우 상대적으로 좋지 않은 수율을 보인다. ARC 방식을 적용한 그림 5의 경우에는 상대적으로 트래픽의 양이 적은 flow 1의 수율을 다소 양보하여, flow 0의 QoS 요구사항을 충족시킴으로써 트래픽의 흐름이 많은 flow 0의 성능이 향상됨을 관찰할 수 있다. 하지만 상대적으로 데이터 트래픽 양이 많은 구간인 10000 슬롯에서 24000 슬롯 사이의 구간에서는 flow 2와 3의 수율이 기존의 스캐터 모드에서의 credit 방식에서 보다 저하됨을 알 수 있다. 비록 다른 링크의 수율의 저하가 다소 있기는 하지만, 보다 많은 트래픽을 갖는 링크에게 보다 많은 서비스 기회를 줄 수 있으므로 관찰된 성능은 타당하다고 볼 수 있다.

다. 하지만 실질적인 데이터의 흐름이 link 0과 link 1 모두 많으므로 두 링크의 수율을 동시에 증가시켜야만 flow 0과 같은 트래픽의 성능을 보장할 수 있다. 따라서 ARC 방식을 적용하여 이러한 데이터 트래픽 별 요구사항을 고려할 경우, 그림 7에서 보는 바와 같이 link 0, link 1 모두 우수한 성능을 나타내었으며, 상대적으로 가중치를 낮게 준 link 2에서의 약간의 성능 저하를 관찰할 수 있다.

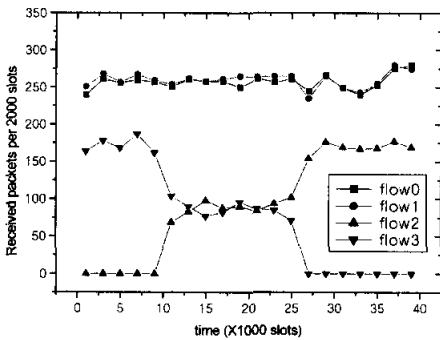


그림 5. ARC 방식을 적용한 스캐터 모드의 서비스 시간에 따른 흐름별 수율 변화.

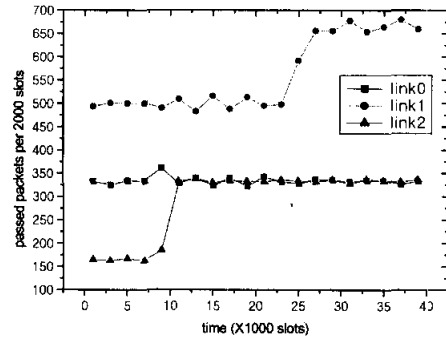


그림 6. 스캐터 모드의 서비스 시간에 따른 링크별 수율 변화(ARC 방식 적용하지 않음).

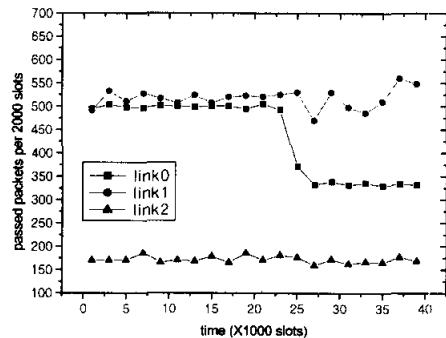


그림 7. ARC 방식을 적용한 스캐터 모드의 서비스 시간에 따른 링크별 수율 변화.

이러한 현상을 링크별로 관찰한 결과를 그림 6과 그림 7에 나타내었다. PMP 노드를 중심으로 각 링크별 수율을 살펴보면, 그림 6에서와 같이 링크별 QoS를 고려치 않은 경우에는 상대적으로 데이터의 흐름이 많으면서 양방향의 흐름을 갖는 link 1의 수율이 가장 우수하고, 데이터 트래픽의 양은 많지만 동일 방향으로의 흐름이 많은 link 0의 성능은 상대적으로 떨어진

#### 4.2 피코넷 스케줄링 알고리즘과의 결합

블루투스의 스케줄링은 크게 두 가지 관점에서 생각할 수 있다. 첫째로 스캐터넷 관점에서 PMP 노드에서의 스케줄링의 시분할 동작을 통

한 스캐터넷 스케줄링과, 둘째로 [4]에서 제안한 알고리즘과 같은 한 피코넷 내에서의 효율적인 자원 분배를 위한 피코넷 스케줄링이 있다. 하지만 이 두 관점은 궁극적인 블루투스의 성능 향상을 위해서는 별개로 동작하여서는 무의미하며, 두 스케줄링이 효과적으로 조화를 이루어 동작할 때에 최적의 성능을 보일 것이다. 따라서 본 절에서는 [4]에서 제안한 차등화 KFP 방식을 앞 절에서 제안한 ARC 방식과 결합하여 전체 스캐터넷 및 각 피코넷에서의 성능을 최적화하고자 한다.

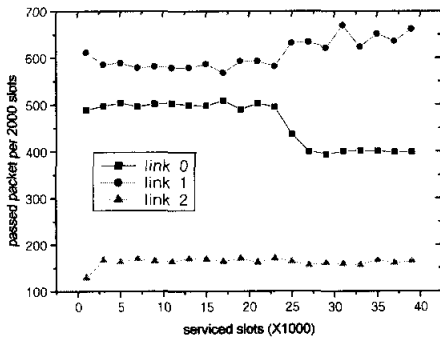


그림 8. ARC 방식과 차등화 KFP 방식을 결합한 경우의 링크별 수율 변화.

그림 8에서 차등화 KFP 방식을 적용한 경우 그림 6과 그림 7의 환경에서의 링크별 성능을 관찰하였다. 그림에서 보듯이, 양방향으로 분산되거나, 트래픽 양이 적은 경우가 아닌 동일 방향으로의 많은 트래픽이 존재하는 link 0에서의 성능 향상이 두드러진다. 즉, 피코넷 스케줄링은 각 링크별, 다시 말해서 피코넷 내부에서의 효율적 동작을 위한 알고리즘이므로 각 링크마다의 스케줄링이 가장 필요한 부분에서 그 효과가 두드러진다고 하겠다. 따라서 그림 9와 같이 흐름별 성능에 있어서는 큰 성능의 향상을 보이지 않는다.

다음으로 차등화 KFP 방식, 즉 피코넷 스케줄링 알고리즘을 적용한 경우의 효과를 보다 구체적으로 알아보겠다. 그림 10 ~ 그림 12에서 그 효과를 나타내었다. 그림 10은 ARC와 차등화 KFP 방식이 적용되지 않은 결과이다. 10000 슬롯 지점에서 flow 2가 발생할 때까지

POLL-NULL 시퀀스로 인한 CE의 중단이 많이 발생하였음을 관찰할 수 있다. 이 경우 슬롯은 버려지거나 피코넷 내부에서 사용될 수 있다. 그러므로 피코넷 스케줄링 알고리즘을 적절히 결합하여 이러한 상황을 줄일 수 있도록 해야 할 것이다. 그림 11에서는 ARC 방식이 적용된 경우이다. 이 경우 PMP 노드에서의 ARC 방식에 의한 효율적인 스케줄링으로 초기의 POLL-NULL 시퀀스 발생을 방지할 수 있었지만, flow 3로 인한 link 0에서의 흐름의 감소가 POLL-NULL 시퀀스의 발생을 다시 유발하였다. 그림 12에서는 이러한 문제점을 극복하기 위하여 차등화 KFP 방식을 그림 11의 경우에 결합하여 스캐터넷만을 고려한 경우에 비해 비교적 성능이 향상됨을 보인다.

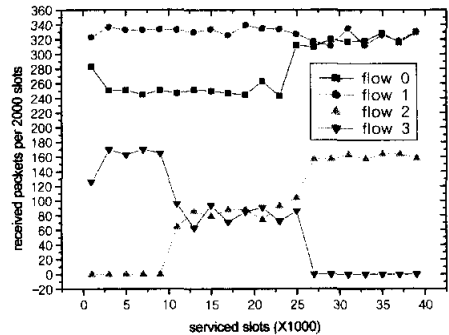


그림 9. ARC 방식과 차등화 KFP 방식을 결합한 경우의 흐름별 수율 변화.

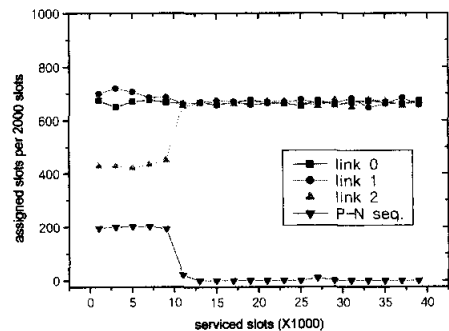


그림 10. 각 링크별 할당된 슬롯(ARC, 차등화 KFP 방식 사용되지 않음).



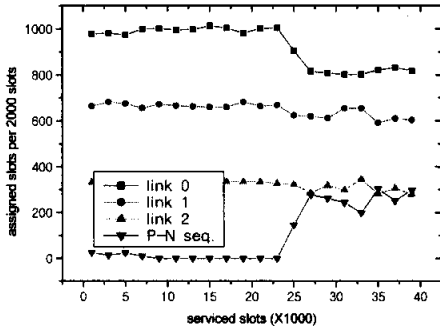


그림 11. ARC 방식이 적용된 경우 각 링크별 할당된 슬롯.

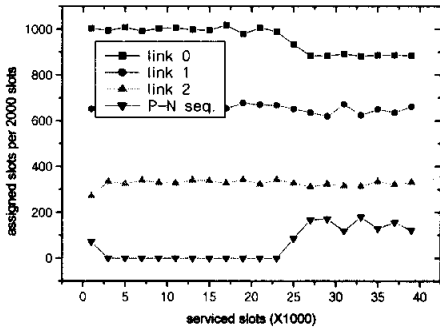


그림 12. ARC와 차등화 KFP 방식이 결합된 경우 각 링크별 할당된 슬롯.

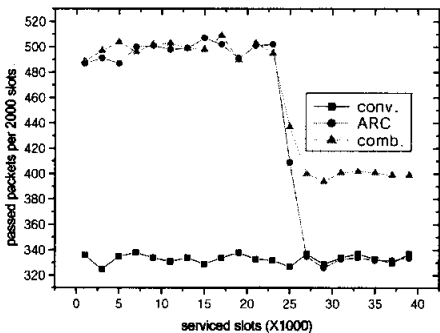


그림 13. link 0에 통과하는 패킷 수.

끝으로 그림 13에서 위의 세 가지 경우에 대하여 link 0에서의 통과하는 패킷의 수를 비교

하였다. 그림에서 보이는 바와 같이 스캐터넷 스케줄링 알고리즘과 피코넷 스케줄링 알고리즘이 결합된 경우에 있어서 그 성능이 가장 우수함을 알 수 있다.

### 5. 결론

본 논문에서는 블루투스 시스템의 스캐터넷 동작의 지원을 위한 효율적인 스케줄링 방안으로 ARC 방식이 제안되었고, 그 성능을 평가하였다. 각 링크별 트래픽 특성에 따른 요구사항을  $T_{scatter\_poll}$ 로 반영하여, 각 피코넷마다 차별화된 서비스를 가능하게 하며, 이를 시뮬레이션을 통해 검증하였다. 또한 제안한 스캐터넷 스케줄링 알고리즘을 기존에 제안한 피코넷 스케줄링 알고리즘과 결합하여 그 성능을 관찰함으로써 보다 향상된 성능을 관찰할 수 있었으며, 이를 통해 피코넷 측면과 스캐터넷 측면에서의 스케줄링 알고리즘의 결합 및 호환이 궁극적으로 효율적인 블루투스 동작을 위해 필요함을 알 수 있었다. 결론적으로 제안된 알고리즘의 적용을 통해 블루투스 시스템의 전반적인 성능을 향상시킬 수 있으며, 본 논문의 결과는 QoS 특성을 지원할 수 있는 블루투스 MAC 스케줄링 방안의 설계에 활용될 수 있을 것이다.

### 참고 문헌

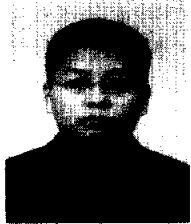
- [1] Bluetooth Special Interest Group, Specification of the Bluetooth System Version 1.1B, Specification Vol. 1&2, Feb. 2001.
- [2] Bluetooth Special Interest Group, Scatter Mode, PAN IPS Whitepaper, Ver. 0.25, June 2002.
- [3] Bluetooth SIG Radio WG [BT PAN IPS], Improvement proposal for Scatternets, Ver. 0.4, Feb. 2002.
- [4] 주양익, 오종수, 권오석, 김용석, 이태진, 엄두섭, 차균현, "블루투스 시스템을 위한 차등화된 형평성 보장 MAC 스케줄링 기법," 한국통신학회 논문지, 심사중.
- [5] S. Baatz, M. Frank, C. Kuhl, P. Martini and C. Scholz, "Adaptive Scatternet Support for Bluetooth using Sniff Mode," in Proc. of

LCN, 2001.

- [6] A. Racz, G. Miklos, F. Kubinszky and A. Valko, "A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets," in Proc. of MobiHOC, pp.193-203, 2001.
- [7] N. Johansson, F. Alriksson and U. Jonsson, "JUMP Mode-A Dynamic Window-based Scheduling Framework for Bluetooth Scatternets," in Proc. of MobiHOC, pp.204-211, 2001.

주 양 익(Yang-Ick Joo)

준회원



1998년 2월: 고려대학교  
전자공학과 학사  
2000년 8월: 고려대학교  
전자공학과 석사  
2000년 9월~현재: 고려대학교  
전자공학과 박사과정

<주관심분야> OFDM, 블루투스, ad hoc 네트워크, 무선 PAN

김 용 석(Yong-Suk Kim)

준회원



1989년 2월: 고려대학교 전자공학과 학사  
1995년 2월: 경기대학교 멀티미디어 통신공학 석사  
2003년 2월~현재: 고려대학교 전자공학과 박사

1989년 3월~현재: 삼성전자

<주관심분야> 블루투스, 무선 PAN/LAN, ad hoc 네트워크

이 태 진(Tae-Jin Lee)

정회원



1989년 2월: 연세대학교 전자공학과 학사  
1991년 2월: 연세대학교 전자공학과 석사  
1995년 12월: University of Michigan, Ann Arbor(M.S.E.)

1999년 5월: University of Texas, Austin(Ph.D.)  
1999년 8월~2001년 2월: 삼성전자 책임연구원  
2001년 3월~현재: 성균관대학교 정보통신공학부 교수

<주관심분야> 통신 네트워크 성능 분석 및 설계, 블루투스, WPAN/WLAN, 광네트워크

차 균 현(Kyun Hyon Tchah)

중신회원



1965년 2월: 서울대학교 전기공학과 학사  
1967년 6월: 미국 일리노이 공과대학 석사  
1976년 6월: 서울대학교 전자공학과 박사  
1977년 3월~현재: 고려대학교

전자공학과 교수

1998년 1월~1998년 12월: 한국통신학회 회장  
2000년 5월~현재: IEE Fellow  
2001년 1월~현재: IEEE Seoul Section Chair

<주관심분야> 블루투스, WPAN/WLAN, 차세대 이동통신 시스템