분산형 블루투스 스캐터넷 형성 프로토콜

정회원 손 진 호*, 정 태 명**

Distributed Bluetooth Scatternet Formation Protocol

Jin-Ho Son*, Tai M. Chung** *Regular Members*

요 약

블루투스 네트워크에서는 여러 피코넷들의 상호 연결을 스캐터넷으로 정의하며, 현재 블루투스 표준에서는 피코넷으로 부터 스캐터넷이 되는 과정을 포함하지 않는다. 기존의 스캐터넷 형성 알고리즘들은 ad-hoc network의 특성을 충분히 반영하지 못하여 노드들의 이동 및 추가, 삭제가 빈번한 시스템에서의 성능 저하를 초래한다. 즉, 스캐터넷의 형성 구조가 복잡해질수록 비효율적인 구성으로 인하여 전송률이 감소하고, 전송 지연이 증가한다. 본 논문에서는 노드들이 분산을 통해 스캐터넷을 형성하는 분산형 블루투스 스캐터넷 형성 알고리즘을 제안하고, 아울러 시뮬레이션 결과를 통하여 제안된 알고리즘이 기존의 알고리즘들보다 우수함을 증명한다.

Key Words : bluetooth, piconet, scatternet, ad-hoc

ABSTRACT

In Bluetooth networks, the scatternet is defined as the internetworking of multiple piconets. Currently, Bluetooth standardization does not include the formation issue of scatternet by piconets. The existing formation algorithms of scatternet do not support the features of ad-hoc networks, which cause the performance degradation of systems when the nodes have certain degree of mobility. Therefore, as the formation of scatternet gets complicated, the throughput is lowered and the delay increases due to the inefficient architectural problems. In this paper, we propose the distributed formation scheme for bluetooth in scatternet, in which the nodes are spread out to form scatternet. Simulation results show that the proposed algorithm outperforms the conventional schemes.

Ⅰ. 서 론

블루투스(Bluetooth)란 핸드폰, PDA, 노트북과 같은 모바일 기기들간의 양방향 근거리 통신을 복잡한 전선 없이 저가격으로 구현하기 위한 표준, 근거리무선통신 기술, 제품을 총칭하여 일컫는다. 이 기술은 2.4 GHz의 비인가 ISM (Industrial, Scientific, and Medical) 주파수 대역을 이용하며, 주파수 도약 확산 스펙트럼(FHSS-frequency hopping spread spectrum) 방식에 기반을 두고 양방향 전송을 위해 시분할 다중 채널 방식(TDD-time division duplex) 방식이 사용된다. 현재 블루투스 표준은 버전 1.1 까지 발표되었으며, IEEE는 2002 년 3 월에 블루투스 표준 1.1의 내용을 IEEE 802.15.1 (WPAN: Wireless Personal Area Network Task Group 1(TG1))의 표준으로 확정하였다[1][2].

블루투스를 이용한 개인영역망은 서로 다른 채널을 사용하는 여러 피코넷간의 연결인 스캐터넷에 기초하며, 스캐터넷은 피코넷과 피코넷을 연결해 주는 브리지 노드 (bridge node)에 의해 형성된다. 이러한 브리지 노드는 어느 한 순간 단 하나의 피코넷 통신에만 참여할 수 있으며, 한쪽 피코넷에서는 마스터 역할을 하고 다른 피코넷에서는 슬레이브 역할을 하는 마스터-슬레이브 브리지(M/S bridge), 혹은 두 피코넷 모두 슬레이브 역할만 하

* LG전자기술원 정보기술연구소 DNT그룹 (jhsohn@lge.com), ** 성균관대학교 전기전자컴퓨터공학과 (tmchung@ece.skku.ac.kr)
논문번호 : 030133-0324, 접수일자 : 2003년 3월 24일

는 슬레이브-슬레이브 브리지(S/S bridge) 로 구분된다. 브리지 노드의 역할과 배치에 따라 각각의 피코넷은 다양한 방법으로 연결될 수 있으며 , 스캐터넷을 구성하는 각각의 피코넷이 연결되는 방법에 따라 다양한 토폴로지 (Topology)의 스캐터넷이 형성될 수 있다. 스캐터넷의 토폴로지는 스캐터넷을 형성하고 유지하는 방법과 시간, 라우팅 방법 등을 결정하므로 스캐터넷의 전체적인 성능과 기능에 영향을 미친다[3].

본 논문 에서는 기존에 제안한 스캐터넷 형성알고리즘의 분석과 ad-hoc network의 이동성으로 고려 한 분산 블루투스 스캐터넷 알고리즘을 제안한다. 이어 지는 2장에서는 블루투스의 스캐터넷 형성에 관한 기존 연구들의 장, 단점을 비교 분석하고, 3장에 서는 알고리즘을 제안한다. 4장에서는 알고리즘의 성능과 시뮬레이션결과를 통한 분석을 보이며, 5장에서는 결론 및 향후 계획에 대해 살펴본다.

II. 관련 연구

1. BTCP

BTCP(Bluetooth Topology Construction Protocol)은 Salonidis, Bhagwat[4][5]등이 제안한 스캐터넷 형성을 위해 리더 선출 기반 알고리즘을 수행 하며, 대칭적 링크 형성 프로토콜(symmetric link formation protocol)에 따라 디바이스들 이 임의적으로 INQUIRY와 INQUIRY SCAN 상태를 교대시킨다. 그리고 투표 과정을 통해 망에 대한 전역적 정보를 가진 유일한 리더를 선출하며, 리더는 슬레이브 노드들을 마스터에게 할당하고 스캐터넷을 구성 할 브리지 노드들을 결정한다. 본 알고리즘은 이후 스캐터넷 형성 알고리즘의 리더 선출 과정에 관한 모태가 되었으나, 연결 단계 (connection phase)와 선출 단계(election phase)가 분리되기 때문에 오버헤드가 크며, 리더가 스캐터넷의 모든 노드들에 대한 정보를 저장하고 있어야 하므로 확장성이 크게 떨어지는 단점이 있다.

2. Bluetree

Bluetree[6] 는 트리(Tree) 구조를 이용한 두 가지 알고리즘을 제시 하였다. 첫번째 알고리즘은 Blueroot라고 하는 하나의 노드를 선택한 뒤 이 노드로부터 각 슬레이브를 새로운 피코넷의 마스터로 추가 하는 방식으로, 남은 슬레이브가 없을 때까지 지속적인 Page 반복 수행을 통해 스캐터넷을 형성하는 방식이다. 두번째 알고리즘은 노드의 역할을 정하고 노드의 수를 2~3 개로 제한하는 방법으로서, 슬레이브의 오버헤드를 줄이는 방식이다. 두 알고리즘 모두 형성하기 간단하고 쉽게 확장할 수 있는 장점을 가지고 있으나, Root 노드를 미리 마스터로 선정하는 문제점과, 정보 수집에 있어서, INQUIRY 시 정보와 PAGE시와의 시간 차이에 따른 노드 이동성에 대한 보장이 불가하다. 또한 링크가 끊어진 경우의 재형성이 어렵고, Tree가 깊어질수록 노드에 많은 부하가 걸린다는 단점을 가지고 있다.

2.3 MIT 연구

Law와 Siu 등이 제안한 블루투스 스캐터넷 형성 알고리즘[7][8]은 스캐터넷 형성의 성능과 위상 효율성(topology efficiency)를 고려하여 최적화된 디바이스 발견 알고리즘 (device discovery algorithm)을 수행함으로써 INQUIRY와 스캐터넷 형성을 위한 시간 복잡도 및 메시지 복잡도를 줄이기 위한 시도를 하였다. 이 알고리즘은 리더 선출 과정에 기반한 알고리즘을 수행하지만, 한번의 단계로 스캐터넷 형성을 완료한다. 하지만 마스터가 INQUIRY 후 접속한 상태에서 스캐터넷을 형성하므로, 지나치게 많은 노드 이동(migration)과정이 생기고, 확장성에서도 많은 제약이 따르며, 용량이 전혀 고려되지 않은 단점이 있다 .

III. 제안 알고리즘

1. 알고리즘의 목적 및 정책

기존의 스캐터넷 형성 프로토콜은 스캐터넷의 형성 과정에서 각 피코넷의 노드 수를 기본적으로 최대 수용 가능한 개수만큼 연결하고 있으며, 이를 위한 다수 불필요한 노드의 이동 (migration)이 발생하게 된다. 이러한 방식의 비효율성은, 한 피코넷에서 슬레이브 수가 3-4개 이내일 경우의 전송률과 지연시간면에서 가장 우수한 성능을 보인 기존 연구의 시뮬레이션 결과로부터 확인할 수 있다. 이러한 사실에 기반하여, 스캐터넷 형성 과정에서 한 피코넷의 노드 수를 3개로 우선 선정하여 높은 전송률과 낮은 전송지연시간을 유지하고, 이후에 접속하는 노드들을 스캐터넷으로 분산시켜 같은 노드

수에 따른 피코넷 형성 보다 용량 (capacity)을 극대화시키면서 리더의 선출과정과 퇴출과정을 최소화 시킨다. 또한 많은 마스터끼리의 접근 시 슬레이브 이동을 최소화하여, 스캐터넷 형성시간을 줄일 수 있다.

제안하는 스캐터넷 형성 알고리즘은 다음의 네 가지 정책을 제시한다.

첫째, 하나의 피코넷 구성에서 슬레이브 수를 가능한 3 개로 제한하여 구성 한다 .

둘째, 3개의 슬레이브로 구성되어있는 피코넷에 접속을 요구하는 노드는 스캐터넷으로 구성한다.

셋째, 피코넷의 최대 접속 노드 수가 7개가 되면 항상 접속되어 있는 피코넷의 노드 수를 먼저 고려하여 리더를 정한다 .

넷째, 브리지는 리더가 될 수 없다 .

리더 선출 과정은 단일 피코넷인 경우와 스캐터넷의 경우로 나뉘어지는데, 전자에서는 마스터가 리더가 되고, 후자에서는 마스터 중 하나가 리더로 선출된다. 또한 마스터 선출 부분에서는, 대부분의 논문에서 그러 하듯이, 스캐터넷 형성 초기에 마스터를 선출할 때 난수를 발생시켜 사용한다. 스캐터넷 형성을 하기 전에 노드들은 각각 리더로서 시작하며, 표 1. Initialize() 함수를 호출한다. 이 함수는 슬레이브 수의 비율 (S(u)/7)보다 작으면 INQUIRY를 수행하고, 이보다 크면 INQUIRY SCAN을 수행하도록 결정한다. 임계 확률 P 는 차후 연구를 통해서 결정하겠지만, 본 알고리즘에서 S(u)/7 의 비율로 구성한 이유는 다음과 같다. 모든 inquiry하는 노드는 같은 호핑 시퀀스(hopping sequence)를 따르기 때문에 패킷 충돌 (packet collision)이 중요한 문제가 될 수 있으므로, 마스터와 슬레이브 비율을 비동기적으로 구성하여 접속률을 높일 수 있기 때문이다. 우선 리더가 된 u가 p보다 적으면 INQUIRY 를 수행하며, v와 접속하게 된 후 표2. CONSTRUCT(u, v) 함수를 호출하게 된다 .
Notation: S(u)는 마스터 u의 슬레이브 개수를 표시 한다.

표 1. Initialize 함수.

```
Initialize(leader u)
1  x ( rand(0, 1)
2  if x <p  //  p= S(u)/7
3        then INQUIRY(u)
4              u connect v          // slave v is founded
5              CONSTRUCT(u, v)
6  else if S(u) = 0
7        then INQUIRY_SCAN(u)
8              PAGE_SCAN(u)    // wait for master
9        else INQUIRY_SCAN(v)
10       PAGE_SCAN(v)    // wait for master
```

표 2. Construct 함수.

```
CONSTRUCT(leader  u, slave v)
1   if v is a leader
2       then switch
3              case  |S(u)|<=3 :    //  그림 1
4              v retire   return
5       case u has no sibling :     //  그림 2
6              y  an unshared slave of u
7              L1_REGISTER(u, v, y)
8              u, v retire y leader return
9       case default :                //  그림 3
10             v retire return
11         else w    the other master of v
12      w retire
13  switch
14      case |S(u)?S(w)|+1<=3  :    //  그림.4
15          MERGE(u, v, w) return
16      case  |S(u)?S(w)|+1<=7 :
17          if u has no sibling      // 그림 5
18          then L2_REGISTER(u, v, w)
19          else MERGE(u, v, w)  // 그림 6
20             return
21      case  default :
22          L2_REGISTER(u, v, w) return
23  if  |S(u)|==7
24        then  bm  ELECT_BEST_MASTER (u)
25          if|S(bm)|==7              //  그림 7
26      then  x, y  an unshared slave of bm
27          L1_REGISTER(bm, x, y)
28            u retire y leader
29      else  u retire bm leader
```

2. 알고리즘 기능 설명

표 2.의 CONSTRUCT 함수는 각 노드들을 연결하는 역할을 수행하는데 단일 노드를 발견하여 접속하는 경우와 노드가 자신의 마스터를 동반하여 접속하는 경우의 두가지를 고려하여 구성된다.

첫째, 노드 v가 단독인 (INQUIRY SCAN)상태에서 이미 피코넷이 형성된 곳으로 접근하여, 다른 피코넷의 마스터 u 가 INQUIRY 를 수행하여 발견하여 접속한 경우

가) u의 슬레이브 수가 3개보다 같거나 적은 경우에는 높은 전송률과 작은 전송지연을 위한 제한된 최대 노드 수를 초과하지 않으므로, 접근한 노드 v 를 u 피코넷의 슬레이브로 만든 후 v 노드가 가진 리더의 역할을 해제함. (그림 1.)

나) u 의 슬레이브 수가 3개보다 많고, u의 형제가 없을 경우에는 제한된 최대 노드 수를 초과하므로, 스캐터넷으로 확장하기 위해 v를 브리지로 하고 , y를 u의 형제로 연결 한다. (그림 2.)

다) 그 외의 모든 경우에는 단순 확장이 용이하므로 , v 를 u 의 슬레이브로 만든다. (그림 3.)

둘째, 슬레이브 v가 마스터 w를 가지고 있는 피코넷에서 다른 피코넷 마스터 u가 INQUIRY를 수행하여 v를 발견하여 접속한 경우

가) u 의 슬레이브 수와 w의 슬레이브 수와 마스터 w 자신의 합이 3개 보다 같거나 적을 경우에는 제한된 최대 노드 수를 초과하지 않으므로, 하나의 피코넷으로 구성하기 위해 w 및 w의 모든 슬레이브들은 u의 슬레이브가 된다. (그림 4.)

나) u 의 슬레이브 수와 w의 슬레이브 수와 마스터 w 의 합이 7개보다 같거나 작을 경우, 최대 노드 수를 초과하므로 하나의 피코넷으로 구성될 수 없으며, 스캐터넷으로 구성되어야 한다. 따라서,

A. u 가 형제가 없는 경우에는, w를 마스터로 한 피코넷을 합쳐 스캐터넷으로 확장하기 위해 , v 가 u 의 브리지인 슬레이브가 된다. (그림 5.)

B. u 가 형제가 있는 경우에는, 이미 스캐터넷으로 구성되어 있으므로 w를 마스터로 한 피코넷은 무시될 필요가 있다. 따라서, w 의 모든 슬레이브와 w 까지 u 의 슬레이브가 된다. (그림 6.)

다) 그 외의 모든 경우 에는 스캐터넷 확장을 위해, v 가 u 의 브리지인 슬레이브가 된다 .

본 함수의 리더 선출 과정을 살펴보면, u의 피코넷에 연결된 모든 피코넷 중에서 슬레이브 수가 가장 적은 피코넷의 마스터가 리더로 선출되는데, 이는 슬레이브 수가 적을수록 연결할 수 있는 노드 수가 많으므로 리더 선/퇴출의 횟수를 줄이고 효율을 높이기 위함이다. 또한, 선출된 마스터 자신도 슬레이브 수가 7 개가 될 때 , 자신의 슬레이브 중 브리지가 아닌 슬레이브(y)를 하나 끊어서 슬레이브(x)와 연결하고 y는 리더로서 선출된다. 이 때 브리지가 아닌 슬레이브를 택하는 이유는 스캐터넷을 형성하고 있는 브리지를 보존함으로써 지속적인 연결을 보장하기 위함이다.(그림 7.)

ELECT_BEST_MASTER 함수는 연결된 모든 피코넷의 슬레이브 수를 검사하여 그 수가 가장 적은 피코넷의 마스터를 선출하는 함수이다.

L1_REGISTER 함수는 u가 v를 발견하여 접속했을 경우 (u자신이 형제 피코넷이 없으며 또는 형제 피코넷이 끊어졌을 경우), u의 슬레이브 중 하나의 슬레이브 를 선택하여 스캐터넷 형태로 접속을 시도하는 함수이 며, L2_REGISTER 함수는 v가 master와 함께 들어온 경우 u의 sibling으로 추가하는 함수이다. 그리고 MERGE 함수는 노드 v가 노드 w로부터 연결을 끊고 노드 v 의 슬레이브와 노드 w자신은 PAGE SCAN 상태, 노드 u 는 PAGE 상태로 각각 전환하여 노드 w 와 노드 w의 모든 slave 가 노드 u의 slave가 되게 하는 함수이다.

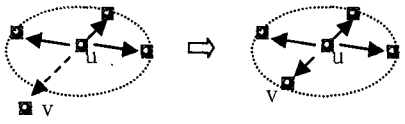


그림 1. CONTRUCT() 함수의 3-4 라인의 예

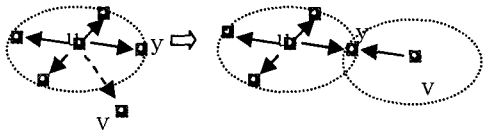


그림 2. CONTRUCT() 함수의 5-8 라인의 예

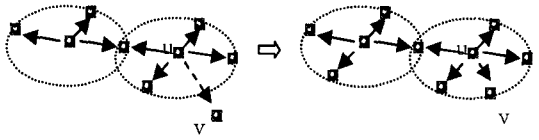


그림 3. CONTRUCT() 함수의 9-10 라인의 예

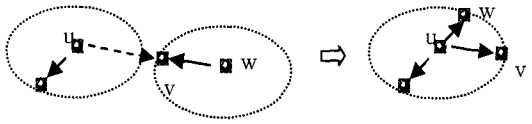


그림 4. CONTRUCT() 함수의 14-15 라인 의 예

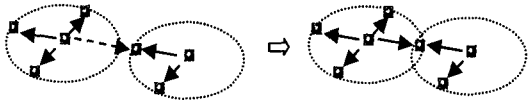


그림 5. CONTRUCT() 함수의 17-18 라인 의 예

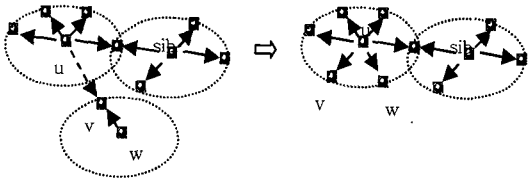


그림 6. CONSTRUCT() 함수의 19-20 라인의 예

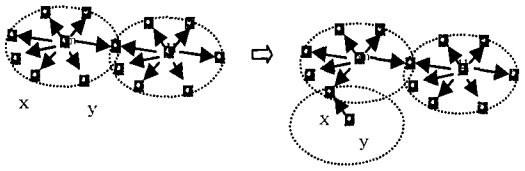


그림 7. CONTRUCT() 함수의 23-29 라인 의 예

Ⅳ. 알고리즘 분석과 시뮬레이션 결과

1.알고리즘 분석

이장에서는 제안한 알고리즘의 상세 분석을 통하여 Cluster 함수의 효율성을 검증한다.
기존의 발표된 논문[7][8] 과 비교하여 분석하면 그림 8,9,10 의 왼쪽에 있는 그림이 Law, Mehta, Siu[8]의 스캐터넷 형성 그림이며, 오른쪽 그림이 제안한 알고리즘의 형성그림이다. 가장 큰 차이점은 , Law, Mehta, Siu[8]의 논문에서는 노드가 하나씩 접속되어 형성될 때 노드의 수는 하나의 피코넷에 최대 6 개까지 접속 가능하지만 , 제안한 논문에서는 노드의 수가 6개가 되기 전에 스캐터넷으로 분산 된다 .

첫 번째 경우 분석 (그림 8.): 어떤 알고리즘이 우수하다고 판단하기 불가능하다. 피코넷의 노드 수를 최대로 수용하면서 네트워크를 구성하느냐 혹은 노드를 분산시켜 스캐터넷으로 구성하느냐는 네트워크 상황에 따라 다르게 고려 될 수 있기 때문이다. 본 연구에서는, 그림 8의 오른쪽 경우처럼 한 개의 브리지인 경우 IV장의 4절의 hold mode를 이용한 스캐터넷 스케쥴링 기법을 통하여 시뮬레이션 하였다 .

두 번째 경우 분석 (그림 9.): Law, Mehta, Siu[8]의 알고리즘은 항상 7 개 되면 반드시 하나를 끊어서 스캐터넷을 대비 하는데, 이미 설명했듯이 이 방법은 한 개의 피코넷의 노드 수가 항상 6 개나 몰려있고 선출된 새 리더는 반드시 마스터 /슬레이브 (M/S)구조로 형성될 수 밖에 없는 비효율적인 구조로 되어있다. 결국 이러한 구조는 브릿지가 마스터로 되어야 만 하는 형태를 갖출 수 밖에 없게 되는데, 이 마스터/슬레이브 구조는 Kalia, Garg, Shorey[9] 논문에서 확인된 것처럼 처리율과 전송지연 측면에서 슬레이브 /슬레이브 (S/S) 구조에 비해 매우 비효율적이다.

세 번째 경우 분석(그림 10.) : 여러 피코넷들이 접속한 경우에는 여러 과정을 거치면서 노드 수가 적은 피코넷으로의 노드 이동이 다수 발생하게 되어 스캐터넷 형성시간이 늘어나고, 따라서 전체 성능이 떨어지게 된다. 이는 애드혹 네트워크에서 많은 시간차를 요구한다는 점을 고려할 때 적합하지 않으며, 이동성 또한 적합하지 않다. 예를 들어 4개의 노드를 가지고 접근하는 두 개의 피코넷의 경우에는 자신의 노드를 3개나 끊고 상대방의 노드에 접속하여 이동시켜야 한다. 그러나 제안논문에서는 노드 수가 적은 초기단계에서부터 노드를 미리 분산시켜 이동을 최소화 했으며, 거의 대칭적인 노드 수를 계속 유지하여 리더의 선/퇴출 없이 분산된 구조를 유지하고 있기 때문에 기존의 노드를 끊고 다른 피코넷으로 이동 할 경우가 줄어든다.

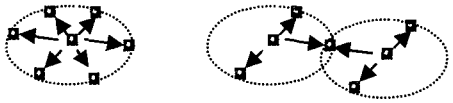


그림 8. 첫 번째 경우의 스캐터넷형성 비교

그림 9. 두 번째 경우의 스캐터넷 형성 비교

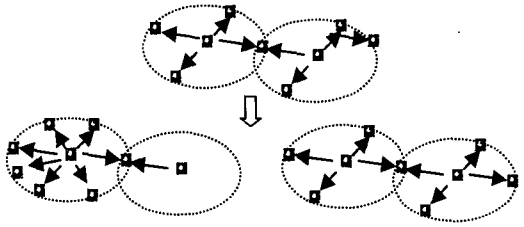


그림 10. 세 번째 경우의 스캐터넷 형성 비교

2. 피코넷 상에서 최적 노드 수 시뮬레이션

피코넷 환경일 경우 최적의 노드수를 판단하기 위해 본 논문에서는 IBM의 블루투스 피코넷 전용 시뮬레이션 툴인 NS기반의 Blue-hoc[10]을 사용하였다. 시뮬레이션은 SCO(Synchronous Connection - Oriented) 채널을 고려하지 않는 순수 ACL (Asynchronous Connectionless) 채널상태만 고려 했

으며, 패킷타입은 DH5 형태만을 고려하 였다. 전달 지연 분석은 마스터에서 슬레이브로 50kbps 의 traffic data 를 슬레이브에게 전송하는 경우로 한정했으며, 스케줄링 기법은 Deficit-round-robin 방법을 이용하였다. 분석을 위한 시뮬레이션 데이터는 다음과 같다. (Error Correction schemes, BER=0.001). 즉, 시뮬레이션 결과 데이터를(그림 11) 분석해 보면, 전송율을 기준으로 볼 때, 노드 수에 따라 최대 이론치인 723kbps 의 속도에서부터 노드 수가 많아질수록 전송률이 떨어지는 것을 알 수 있다. 전송률은 선형 (Throughput/n, n=노드수) 에 가까운 형태로 떨어지며, 지연 은 노드 수가 4 개가 되면서부터 늘어나고 있음을 알 수 있다.

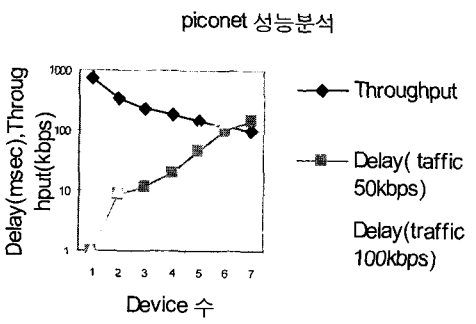


그림 11. 피코넷상에서 적합한 슬레이브수

3. 분산 스캐터넷 알고리즘 시뮬레이션 결과

이 절에서는 본 논문의 성능에 영향을 미칠 파라미터로 다음과 같은 파라미터를 설정하였다.

- 각 라운드 수에 따른 피코넷 수
- 각 라운드에 따른 피코넷 당 평균 슬레이브수
- 연결을 위해 마스터와 슬레이브 접속을 끊는 횟수

스캐터넷 형성단계는 본 논문의 알고리즘과 기존의 Law, Mehta, Siu 가 제안한 스캐터넷 알고리즘[10], 그리고 노드가 무작위로 연결되는 Basagni, Bruno, Petrioli의 알고리즘[4]에 대해, 피코넷의 수와 피코넷 당 평균 슬레이브 수를 각 라운드에 대해 비교하였으며, 아울러 리더의 퇴출 회수와 형성이 완료되기 위한 총 라운드 수를 측정하였다.

스캐터넷 형성을 위한 시뮬레이션은 총 30 개의 디바이스에 대하여 1000 회의 반복 수행을 통해 검증 결과를 얻었으며, 각 알고리즘의 수행 라운드에 대한 피코넷 수와 피코넷 당 평균 슬레이브 수를 나타낸 그래프는 각각 그림 12.와 그림 13. 과 같다. 우선 , 그림 12. 의 그래프의 결과를 보면 , 제안 알고리즘의 피코넷 수가 각 라운드에 대해 대체로 균등하게 분포함을 알 수 있으며, 이는 다른 스캐터넷 형성 알고리즘에 비해 노드의 이동(migration)이 적다는 사실을 유추할 수 있게 한다. 그리고 그림 13.의 그래프에서는 초기 수행 라운드에서 타 알고리즘에 비해 평균 슬레이브 수가 적고 균등하므로 스캐터넷의 전송률 및 지연시간의 측면에서 보다 우수한 성능을 보장할 수 있게된다 . 본 시뮬레이션에서 30개의 노드가 하나의 스캐터넷으로 연결되는데 요구되는 평균 라운드 수는 제안 알고리즘이 8.144 회 , Law 등의 알고리즘[10]이 8.077 회, Basagni 등의 알고리즘[4]이 7.176 회로 측정되었으며, 리더의 퇴출 회수는 제안 알고리즘이 29.085 회, Law 등의 알고리즘이 28.204 회로 측정되어 대체로 유사한 결과를 보였다 .

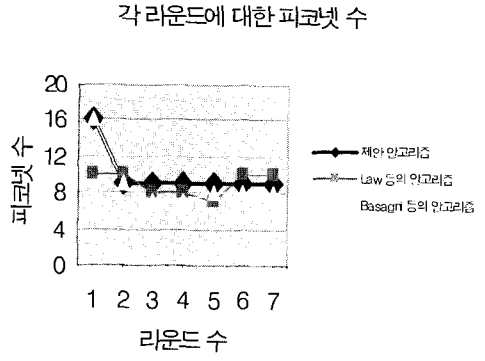


그림 12. 각 라운드에 대한 피코넷 수

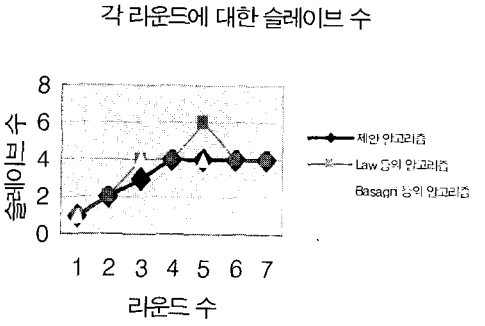


그림 13. 각 라운드에 대한 평균 피코넷 당 슬레이브 수

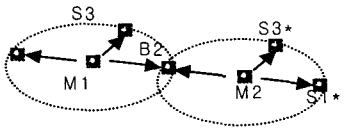
그 밖에도 두 개의 피코넷을 서로 연결하기 위해 마스터와 슬레이브의 접속을 끊어야 하는 회수가 제안 알고리즘은 평균 25.229 회임에 반해 , Law 등의 알고리즘은 25.583회로 제안 알고리즘의 노드 이동 및 접속 해제 회수가 적다는 사실을 뒷받침 해주고 있다.

4. Hold mode를 이용한 Propagation Delay 을 검증

블루투스 표준[1]에서부터 [Erratum 1188 참조] bandwidth 의 낭비를 막기 위하여 Hold 명령이 전달되는 순간에 hold_instant parameter를 규정하고 있으며, 또한 hold_instant의 시간을 최소한 6*Tpoll 보다 크도록 정의하고 있다. 이는 전송하는 디바이스의 Link Controller 가 hold_instant 값을 LMP_hold 메시지를 통하여 전송할 때, 전송받는 디바이스에서 메시지를 받은 순간의 hold_instant 값이 이미 경과했을 경우를 고려한 것이다. 또한 슬레이브가 Hold mode 상태에서 다시 돌아와서 동기화를 맞추는 시간은 최소 500 microsecond와 주어진 Uncertainty window 사이즈를 고려하여 +/- 510 microsecond 가 필요하며, 이때 필요한 slot의 수는 슬레이브의 LMP_timming_accuracy_req 와 새로운 마스터의 LMP_timming_res 를 고려하여 최소 2 slot이 필요한 것으로 분석된다. 결론적으로 스캐터넷 형성 후 전파되는 시간은 다음과 같이 정량화 할 수 있다 .

6*Tpoll + 2*synchronizing + slave 수 에 따른 time slot수

결국 Law, Mehta, Siu[9] 논문과 제안한 논문의 스캐터넷 형성을 비교해보면 마스터가 라운드로빈 방식으로 폴링할경우 크게 두 가지 형태 즉, 그림. 14의 S3에서 S3*로 데이터를 전송하는 경우와 그림 15. 의 S3에서 S2로 데이터를 전송하는 경우로 구분될 수 있다. 또한, 최악의 경우를 포함하여 각 노드의 최소 전송 지연시간과 최대 전송 지연시간을 평균값으로 계산하면 최대 4.6 slot의 차이를 보인다. 즉 625ms x 5slot 만큼 지연됨을 알 수 있다.



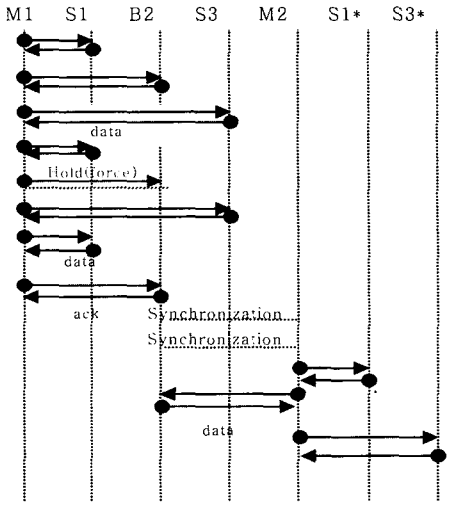
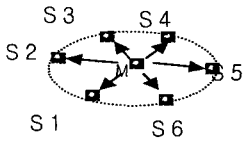


그림. 14. S3 에서부터 S3* 까지의 전송지연측정



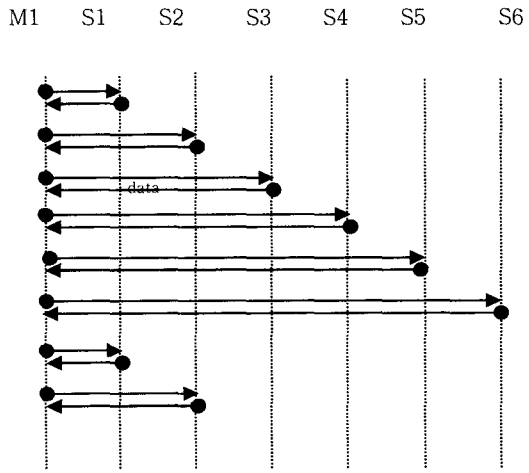


그림 15. S3에서부터 S2 까지의 전송 지연 측정

V. 결 론

본 논문에서는 기존의 스캐터넷 형성 알고리즘의 성능을 비교 분석해보고 이를 토대로 ad-hoc network의 특성을 고려한 분산형 스캐터넷 형성 알고리즘을 제안하였다. 아울러 시뮬레이션을 통해 제안한 형성 알고리즘의 성능을 평가하고 타당성을 검증해보았다 .

그 결과로 전체적인 성능은 스캐터넷 형성 단계에서 Law, Siu 등의 제안한 알고리즘[7][8] 과 동일한 수준의 스캐터넷 형성 시간을 유지하면서, 피코넷 당 평균 슬레이브 수가 적게 구성됨으로써 전반적인 스캐터넷 전송률을 높이고 지연시간을 줄였으며, 따라서 제안된 알고리즘의 스캐터넷 성능이 우수함을 검증 하였다. 또한 제안 알고리즘은 피코넷 당 평균 슬레이브 수를 적게 유지하기 때문에 스캐터넷의 평균 피코넷수, 즉 스캐터넷의 홉수가 다른 알고리즘에 비해 많음을 볼 수 있었다. 그러나 이 부분도 스캐터넷의 형성을 위한 라운드 수가 증가함에 따라 피코넷수도 기존의 알고리즘의 피코넷수와 비슷한 수준을 유지해 간다는 사실을 시뮬레이션으로부터 확인할 수 있었다. 즉, 홉 수의 증가는 본 논문의 IV장의 4절에서 살펴본 것처럼, 피코넷과 스캐터넷의 구성에 따른 전송 지연이 최대 625 ms x 5slot 만큼 늘어날 것으로 보인다. 이는 상황에 따라 성능의 저하를 가져올 수 있으나, 노드의 이동이나 형성시간이 줄어드는 장점과 스캐터넷으로 구성된 후부터는 높은 전송율과 낮은 지연시간을 최대한 유지하므로 전체적인 성능측면에 오히려 우수하다고 판단된다.

본 논문의 향후 과제는 제시된 알고리즘 을 토대로 하여 실제 블루투스 스택에 알고리즘을 적용할 예정이다 .

참 고 문 헌

[1] Spec. of the Bluetooth System Version 1.1, Bluetooth Special Interest Group, http://www.bluetooth.org

[2] Jaap Haartsen, ERICCION Radio System B. V. "The Bluetooth Radio System", *Personal Communications IEEE*, 2000

[3] Jennifer Bray, Charles F. Sturman, "Bluetooth 1.1 Connection Without Cables" Second Edition, *Prentice Hall PTR.*

[4] Theodoros Saloni dis , Pravin Bhagwat, Leandros Tassiulas, Richard LaMaire "Distributed topology construction of bluetooth personal area networks", *INFOCOM IEEE* 2001.

[5] Theodoros Salonidis, Pravin Bhagwat, Leandros Tassiulas, "Proximity awareness and fast Connection establishment in Bluetooth", *In First Annual Workshop on Mobile and Ad Hoc Networking and computing,* 2000

[6] Zaruba G.V., Stefano Basagni, Imrich Chlamtac, "Bluetrees Scatternet formation to enable bluetooth-based ad hoc networks", *IEEE Communications,* 2001

[7] Ching Law , Kai-Yeung Siu, MIT "A Bluetooth Scatternet Formation Algorithm," *IEEE Symposium on Ad Hoc Wireless Networks,* 2001

[8] Ching Law, Amar K. Mehta, Kai-Yeung Siu, MIT, "Performance of A New Bluetooth Scatternet Formation Protocol", *IEEE,* 2001

[9] Manish Kalia, Sumit Garg, Rajeev Shorey IBM India Research lab," Scatternet Structure and Inter-Piconet Communication in Bluetooth System", *Technical Report,* 2000

[10] Bluehoc : Bluetooth performance evaluation tool. http://oss. software .ibm.com/developerworks/open souece/bluehoc/

손 진 호(Jin-Ho Son) 정회원

1993년 3월~현재 LG전자기술원 책임연구원
1991년 2월 : 성균관대학교 학사
1993년 2월 : 성균관대학교 석사
1999년 3월~현재:성균관대학교 전기전자컴퓨터공학과 박사과정

<주관심분야>SDR, WPAN(Bluetooth zigbee), Home Network, Middleware(UPnP, Jini)

정 태 명(Tai M. Chung) 정회원

1995년 8월~현재 성균관대학교 교수
1981년 2월 : 연세대학교 학사
1984년 2월 : University of Illinois at Chicago 학사
1984년 2월 : University of Illinois at Chicago 석사 .
1995년 8월: Electrical and Computer Engineering Purdue University 박사 .

<주관심분야>Ad-hoc Network, NMS, IDS, VPN, Grid Security, Active Network Tai M. Chung