
모바일 환경에서 블루투스 스캐터넷 형성 알고리즘 분석

이대봉* · 장종욱*

Analysis of Bluetooth Scatternet Formation Algorithm over Mobile Networks

Dae-bong Lee* · Jong-wook Jang*

요 약

일반적으로 블루투스 기술은 기기들이 하나의 중심기기에 7개의 보조기기가 연결된 소형 네트워크를 피코넷 구성하게 하는 통신규약을 의미한다. 스캐터넷은 피코넷과 피코넷 간의 통신으로 이루어져 있다. 스캐터넷의 형태로는 트리와 링 형태가 있다. 그리고 링 형태가 모바일 환경에 더욱 적합하다고 한다. 본 논문은 트리와 링 구조의 성능을 비교 분석하여 어떤 형태가 모바일 환경에 적합한지를 알아보았다. 성능 측정 은 단말기들이 빈번하게 추가 삭제되는 모바일 환경에서 스캐터넷 형성 알고리즘을 분석하여 스캐터넷 형성시간을 비교하였다. 실험 결과는 노드 수의 증가에 따라 트리 구조보다는 링 구조가 스캐터넷 형성 시간이 빠르고 시간의 차이의 폭이 넓어졌다.

ABSTRACT

A multi-hop ad-hoc network can be formed out of Bluetooth capable devices through the use of scatternets. Many topologies can be deployed for scatternet; the most commonly proposed topology in the literature are tree structure and ring structure. We analyze the performance between tree formation algorithm and ring formation algorithm. We then present quantitative result that compare the expected traffic performance between Bluetooth ring scatternet and Bluetooth tree scatternet. The simulation results shows that Ring structure of scatternet have good performance than Tree structure over mobile environments.

키워드

Bluetooth, Scatternet, Mobile, Piconet

1. 서 론

개인용 컴퓨터 시장이 침체되는 경향을 보이는 상황에서 이동전화, PDA, 노트북 등을 이용한 이동 컴퓨터는 점점 더 많은 관심을 받고 있다. 그러나 이러한 단말기들은 호환성을 가진 데이터 통신 인터페이스를 보유하고 있지 않으며, 있더라도 번거로운 케이블 연결과 설치 절차가 필요하다. 이러한 상황에서 가장 합리적인 방법은 편리한 단말기들 간의 연결을

위해 선을 제거하고 근거리 무선통신을 사용하는 것이다^[1,2].

블루투스는 이러한 환경을 위해 저렴한 비용으로 모바일 디바이스를 연결할 수 있도록 설계되었고 현재 블루투스 규격 버전1.1은 무선 통신 인터페이스, 관련 통신 프로토콜과 사용 프로파일을 정의하고 있으며, 블루투스의 전송 속도, 통신 반경, 저비용, 저전력, 단일 칩 구현을 위해 설정되었다. 따라서 좁은 범위의 무선 ad-hoc 네트워크를 구성하기 위한 유력

한 기술로 떠오르고 있다^[2].

대부분의 무선 ad-hoc 네트워크 기술들은 단일 채널의 방송 기반(Broadcast based) 방식에 바탕을 두고 있다. 따라서 네트워크의 모든 노드들이 CSMA방식의 MAC 프로토콜을 통해 하나의 채널을 공유하게 되어 네트워크의 토폴로지가 메쉬형(Mesh Topology) 하나로 결정된다. 그러나 블루투스의 경우 주파수 호핑(Frequency Hopping)을 기반으로 하는 다중 채널 방식에 바탕을 두고 있기 때문에 각각의 채널을 연결하는 방식에 따라 다양한 형태의 네트워크가 형성될 수 있다.

일반적으로 블루투스는 하나의 마스터 기기에 최대 7개의 슬레이브 기기가 연결된 소형 네트워크를 피코넷이라 한다. 이러한 피코넷이 모여서 이루어진 것이 스캐터넷이다.

본 논문에서는 블루투스의 기술이 적용되는 주요 대상이 모바일 디바이스라는 점을 감안하여 스캐터넷 형성 과정에서 모바일 환경을 고려한 스캐터넷 알고리즘을 구현하였다. 스캐터넷의 네트워크 토폴로지는 여러 가지 형태가 있지만 가장 보편적으로 사용되는 트리 토폴로지와 링 토폴로지에 초점을 맞추었다. 일반적으로 트리 토폴로지로 네트워크를 구성하여 성능을 분석한 연구는 많았다. 하지만 트리 토폴로지가 모바일 환경에서 다른 토폴로지보다 성능이 뛰어난지는 알 수 없었다. 그래서 링 토폴로지의 네트워크와 비교하기 위해서 스캐터넷의 형성 시 두 가지 토폴로지로 스캐터넷 형성 시간을 구하여 비교하였다. 그리하여 두 가지 토폴로지 중 어떤 것이 모바일 환경에 적합한가를 알아보기로 한다.

본 논문의 구성은 2장에서 블루투스의 개요에 대한 설명하고, 3장은 블루투스 네트워크에서 피코넷과 스캐터넷에 대해 알아본다. 4장에서는 스캐터넷 형성 알고리즘으로 트리와 링 토폴로지로 형성되는 과정을 설명하고, 5장은 모바일 환경에서 블루투스 스캐터넷 형성 알고리즘 분석을 위한 시뮬레이션하여 스캐터넷 형성시간을 측정하였다. 마지막으로 6장에서 결론으로 끝을 맺는다.

II. 블루투스 프로토콜

블루투스 규격은 블루투스 SIG(Special Interest Group)에서 제정되며 코어 부분과 프로파일 부분으로 구분된다. 코어 규격은 블루투스 프로토콜 스택의 모든 계층을 정의하며, 이는 그림 1.과 같다.

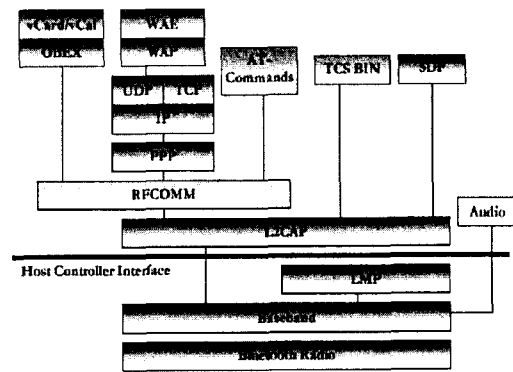


그림 1. 블루투스 프로토콜 스택
Fig. 1 Bluetooth Protocol Stack

스택의 가장 아래에 있는 라디오 계층의 규격은 무선 특성, 주파수 대역, 채널 할당, 전송 파워, 수신 감도를 정의한다. Baseband 계층은 블루투스의 물리적 계층과 MAC(Medium Access Control) 계층을 담당하며 단말기 발견(device discovery), 링크 형성, 단말기 간의 동기-비동기 통신과 같은 일을 수행한다. 블루투스는 단말기 간에 baseband 연결 관리를 위해 여러 제어 메시지를 주고받아야 하며, 이러한 메시지들의 정의는 LMP(Link Manager Protocol)의 일부분이다. LMP와 관련된 일들을 수행하는 기능적 엔티티(entity)를 링크 관리자라 한다^[3,5].

L2CAP(Logical Link Control and Adaptation Protocol) 계층은 블루투스의 링크 계층에 속한다. L2CAP은 상위 계층으로부터 받은 패킷을 L2CAP 연결을 통해 전송하며 상위 계층의 다중화와 분할(segmentation), 조합(reassembly) 기능을 수행한다. SDP(Service Discovery Protocol) 계층 규격은 클라이언트 단말기 서비스와 그 서비스에 대한 특성을 발견하는 방법을 정의한다.

RFCOMM 계층 규격은 RS-232 인터페이스를 블루투스 무선 링크 상에서 에뮬레이션 하는 방법을 정의한다. RFCOMM은 피어(peer) 단말기와 통신기를 위해 COM 포트를 사용하는 전통적인 애플리케이션을 지원한다^[7].

III. 스캐터넷 형성

3.1 트리 스캐터넷 형성

트리 구조는 스캐터넷 형성에서 처음에 노드 하나가 초기에 주어진다. 이 노드는 트리 구조에서 루트에 위치하는 노드를 Blueroot라고 부른다. 여기서 루트 노드는 마스터가 되고 주위의 한 홉 안에 있는 노드는 피코넷에서의 슬레이브 노드가 된다^[4]. 즉, 루트 노드의 자식 노드가 된다. 자식 노드의 이웃에 새로운 노드가 존재하면 자식 노드는 마스터가 되고, 그렇지 않으면 leaf 노드가 되어 단지 슬레이브 노드가 된다^[5]. 이러한 방식으로 네트워크가 구성되어진다.

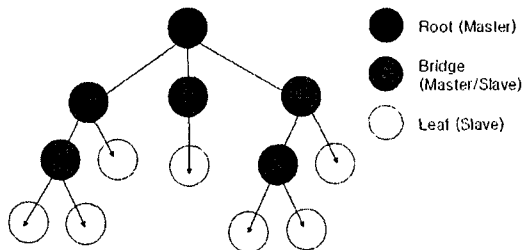


그림 2. 블루트리 스캐터넷
Fig. 2 Bluetree Scatternet

그림 2.는 루트 노드를 중심으로 형성된 트리 스캐터넷을 보여주고 있다. 루트 노드는 항상 마스터인 상태이다. 그리고 노드와 노드를 연결하는 브릿지 노드로 이 노드는 마스터와 슬레이브 역할을 바꾸어가면서 브릿지 역할을 한다. 마지막으로 트리에서의 가장 끝단에 있는 leaf 노드는 단지 슬레이브 역할만 한다. 만약 leaf 노드의 하위에서 새로운 노드가 생성되게 되면 leaf 노드는 브릿지 노드로 바뀌게 되면서 마스터/슬레이브의 두 가지 역할을 하게 된다.

3.2 링 스캐터넷 형성

링 구조에서 구조의 주축이 되는 노드는 마스터/슬레이브 브리지가 되며 각 마스터/슬레이브 노드에 leaf 노드인 슬레이브 노드가 연결되어 구성되어진 피코넷이다^[6]. 링 스캐터넷은 이러한 피코넷들을 링 형태로 네트워크를 구성하게 된다. 링 형태의 스캐터넷은 조정자(Coordination) 노드를 중심으로 형성되며, 또한 조정자 노드는 노드의 추가 삭제에 따른 재구성을 담당한다.

그림 3.은 링 스캐터넷 구성을 보여주고 있다. 그림에서 확인 할 수 있듯이 조정자 노드가 한 피코넷의 마스터가 되어 피코넷을 형성한다. 그리고 브릿지 노드는 조정자 역할을 거친 노드들이다. 브릿지 노드를 링 형태로 만들로 중심에 조정자가 존재하여 모든 브릿지 노드를 관리하게 된다.

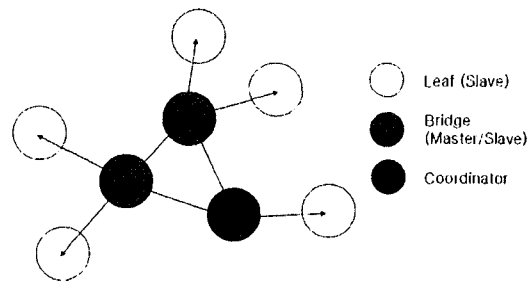


그림 3. 블루링 스캐터넷
Fig. 3 BlueRing Scatternet

IV. 모바일 환경에서 블루투스 스캐터넷 형성 알고리즘 분석을 위한 시뮬레이션

4.1 시뮬레이션 파라미터 및 시나리오

본 논문에서 3장에서 제시한 트리와 링 토폴로지를 각각 구현하여 성능을 분석하기 위해서 모바일 환경과 노드들의 시간에 따른 시나리오를 만들었다. 시뮬레이션 파라미터는 다음과 같다.

- 네트워크 크기는 100m X 100m
- 노드의 개수가 10, 15, 20, 25, 30개 일 때 재구성 시간의 평균 측정

- 피코넷의 슬레이브 개수를 3개로 구성
- 대역폭 1Mhz, 슬롯 길이 625 μ s
- 전파영역 10m

모바일 환경에서의 스캐터넷을 구현하기 위해서 공개 시뮬레이터인 NS-2를 이용하였다. 이는 NS-2 시뮬레이터에서 모바일 노드를 지원해 주기 때문이다. 또한 IBM에서 개방한 Bluehoc도 NS-2 기반으로 구현되었기 때문에 결과에 대한 신뢰성도 어느 정도 보장이 가능하다.

스캐터넷의 형성 알고리즘을 비교 분석하기 위해서는 기본적으로 트리나 링 형태의 스캐터넷이 구성되지만 모든 노드들의 이동 시나리오는 동일한 환경을 적용하였다.

먼저 노드들의 시작위치를 고정시켜서 트리나 링 스캐터넷을 만들게 하였다. 그림 4.는 시뮬레이션을 애니메이션으로 보여주는 NAM 프로그램을 사용하여 노드들의 시작위치를 나타내었다.

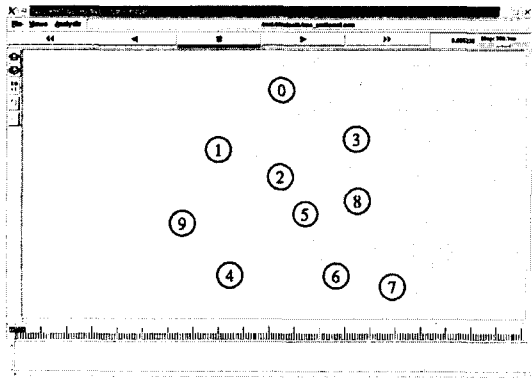


그림 4. 모바일 노드의 시작 위치
Fig. 4 Start location of Mobile node

Tcl 언어로 노드의 위치를 선정할 때 3차원으로 위치를 정할 수 있다. 그러나 본 논문에서는 2차원에서의 좌표만을 주어 노드를 이동 시켰다.

```
# 이동 노드들의 초기 위치
$node_(0) set X_ 400.0
$node_(0) set Y_ 250.0
$node_(0) set Z_ 0.0
...
$node_(9) set X_ 321.0
$node_(9) set Y_ 142.0
$node_(9) set Z_ 0.0
```

시뮬레이션에서 가장 중점을 둔 것이 노드들이 이동 단말기라는 것과 노드들의 이동성이 보장된다. 때문에 노드의 추가와 삭제는 노드의 이동에 따른 전파영역의 범위를 벗어나면 노드는 삭제와 같은 역할이 되고 새로운 노드가 들어오면 노드의 추가와 같은 역할을 하게 된다. 따라서 노드들의 이동 시나리오 작성 시 이러한 환경을 고려하여 만들었다.

노드들의 초기 위치에서 시뮬레이션 실행 시 0.0 초에서 노드들은 각각 목적지 위치로 이동하기 시작한다. 그리고 목적 위치까지 도달하게 되면 멈추게 된다. 그리고 60 초마다 노드들은 다른 목적지를 향해 이동하게 된다. 이때 노드는 전파영역의 범위를 벗어나게 된다. 그리고 전파영역을 벗어난 노드는 다른 영역으로 가게 되어 재구성된다. 이렇게 300초만에 노드의 이동을 빈번하게 시나리오를 작성하였다.

```
# 0.0초에 노드들이 이동 함
$ns_ at 0.0 "$node_(1) setdest 320.0 240.0 0.0"
$ns_ at 0.0 "$node_(0) setdest 480.0 180.0 0.0"
...
$ns_ at 0.0 "$node_(9) setdest 260.0 135.0 0.0"

# 60.0초에 노드들이 이동 함
$ns_ at 60.0 "$node_(1) setdest 220.0 140.0 0.0"
$ns_ at 60.0 "$node_(0) setdest 380.0 80.0 0.0"
...
$ns_ at 60.0 "$node_(9) setdest 160.0 335.0 0.0"
...
```

4.2 시뮬레이션

트리와 링 형태의 형성시간을 비교하기 위해 알고리즘을 구현하여 시뮬레이션을 하여 보았다. 트리 형태에서 루트노드를 0번으로 설정하여 트리 형태를

형성하게 하였고, 링 형태도 마찬가지로 0번 노드를 시작 노드로 설정하였다.

트리 형태의 형성을 보면 루트노드인 0번을 중심으로 트리 모양으로 형성되었다. 스캐터넷을 형성 결과 그림 5와 같이 나타났다. 쉽게 알아보기 위해서 형성된 트리 스캐터넷에 노드간의 링크를 실선을 그어 나타낸 것이다.

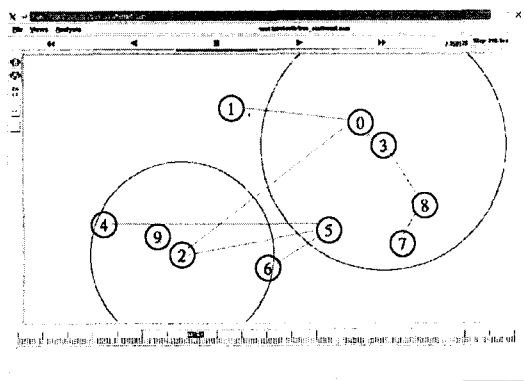


그림 5. 트리 스캐터넷 형성
Fig. 5 Formation of Tree Scatternet

링 형태에서의 형성은 조정자 노드인 0번을 시작으로 링 모양으로 형성되었다. 스캐터넷 형성 결과는 그림 6과 같이 형성되었다.

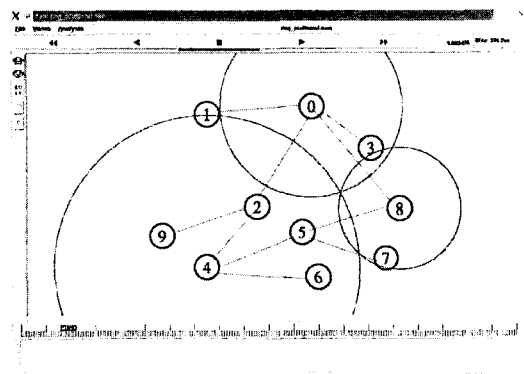


그림 6. 링 스캐터넷 형성
Fig. 6 Formation of Ring Scatternet

형성된 스캐터넷은 데이터 패킷을 전송하기 시작하는데 이때 사용된 프로토콜은 DSR (Dynamic Source Routing) 프로토콜을 사용하였다. 이 프로토콜은 소스 라우팅 개념에 근거한 reactive 라우팅 프로토콜로서, 1996년 Carnegie Mellon 대학의 David B. Johnson과 David A. Maltz에 의해 개발되었다 [8]. 단말기들은 경로 저장을 위한 route cache를 보유해야 하며, 새로운 경로를 발견하면 지속적으로 정보를 갱신한다.

4.3 시뮬레이션 결과 분석

스캐터넷의 트리와 링 토폴로지에서의 형성 시간을 측정하여 평균 시간을 구하였다. 그래서 노드 수에 따른 토폴로지 형성 시간이 그림 7과 같이 결과가 나왔다. 그림 7에서 노드수가 작을 때는 트리나 링 토폴로지가 거의 시간의 차이가 나지 않는다. 그러나 네트워크에 노드수가 늘어감에 따라 형성 시간의 폭이 크다는 것을 알 수 있다. 따라서 네트워크에서 이동 노드의 증가에 따라 링 토폴로지가 트리 토폴로지보다 짧은 형성 시간을 가진다.

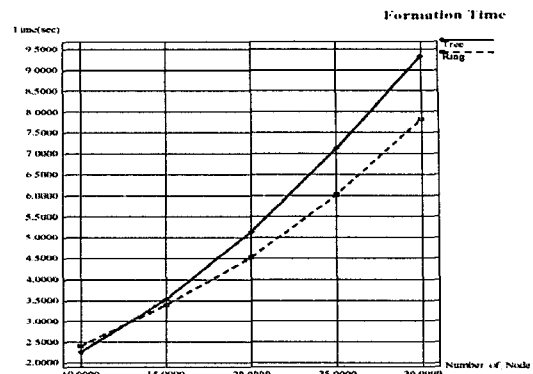


그림 7. 스캐터넷 형성 시간
Fig. 7 Scatternet Formation Time

트리와 링 토폴로지에서의 패킷의 전송시간을 측정하였다. 두 토폴로지를 비교하기 위해서 같은 홉 길이와 추출한 패킷의 수를 같게 하였다. 그래서 추출된 패킷의 전송시간에 대해 평균 시간을 구하여 그림 8과 같이 결과가 나왔다. 그림 8에서 트리 토폴로지 보다는 링 토폴로지가 더 짧은 시간에 패킷

을 전송함을 알 수 있다.

네트워크의 토폴로지에 따른 스캐터넷 형성 시간을 트리 토폴로지와 링 토폴로지로 구분하여 시간을 측정하였고, 패킷의 전송시간도 함께 알아 측정하여 알아보았다.

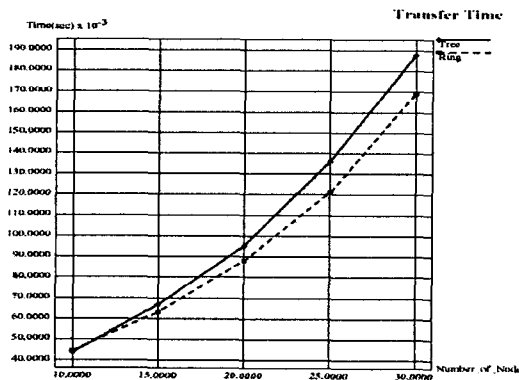


그림 8. 패킷 전송 시간
Fig. 8 Packet Transfer Time

그림 7.에서는 토폴로지에 따른 스캐터넷의 형성 시간을 네트워크의 노드 수(10개, 15개, 20개, 25개, 30개)의 증가에 따라 시간을 측정 하여 보았다. 그 결과 노드수가 10개와 15개 사이에는 시간이 비슷하게 측정이 되었다. 그러나 노드수가 20개에서 점차 늘어남에 따라서 트리와 링 토폴로지의 형성 시간의 차이가 커져 갔다. 그림 8.은 패킷의 전송 시간을 측정 한 것으로서 이는 토폴로지에 따른 패킷의 전송 시간이 어떻게 변하는가를 알아보기 위해서이다. 결과 링 토폴로지가 노드수가 많아짐에 따라 짧은 전송시간을 가짐을 알 수 있다 그러나 전송시간의 차이는 형성 시간의 차이보다 훨씬 적기 때문에 크게 차이는 나지 않는다.

여기서 알 수 있듯이 노드 수가 적을 때는 트리 토폴로지나 링 토폴로지나 별 차이를 알 수 없지만 네트워크 상에서 노드수가 증가함으로 인해 링 토폴로지가 트리 토폴로지 보다 짧은 시간을 가진다. 따라서 모바일 환경에서 스캐터넷의 토폴로지는 링 토폴로지가 더 좋은 토폴로지임을 보여 주고 있다.

V. 결 론

현재 스캐터넷 설정에 대한 내용은 아직까지도 명확하게 SIG에서 규정해놓지 않은 상태이다. 그러나 현재 모바일 디바이스가 늘어가는 추세로 볼 때 블루투스에서 스캐터넷을 빼고는 큰 발전을 생각할 수 없다. 그래서 본 논문에서는 모바일 환경에서 블루투스 스캐터넷 형성에서 트리와 링 토폴로지를 구현하여 성능을 비교 분석하였다.

스캐터넷 형성 시간을 알아보기 위해 트리 토폴로지와 링 토폴로지에서 노드수를 10개로 하고 형성 시간을 측정한 결과 링보다는 트리 토폴로지가 낮은 형성 시간을 보였다. 그러나 모바일 환경에서 이동 노드의 수를 15개, 20개, 25개, 30개로 점차 늘려서 형성 시간을 측정해 보았다. 그 결과 트리와 링 형성 시간의 차이는 0.08, 0.65, 1.12, 1.62초로 차이가 늘어남을 알 수 있다. 즉, 노드 수가 증가함에 따라서 링 토폴로지가 트리 토폴로지 보다 낮은 형성 시간을 보였고 형성 시간의 차이는 노드 수가 증가함에 따라 큰 폭으로 늘어났다.

패킷의 전송시간이 토폴로지에 따라 어떻게 변하는지를 알아보기 위해서 링과 트리 토폴로지 전송시간을 구하였다. 우선 노드수가 10개 일 때 패킷의 전송시간을 구한 결과 거의 차이가 나지 않았다. 그래서 이것도 마찬가지로 노드수를 증가시키면서 전송시간을 구하여 보았다 그 결과 링 토폴로지가 트리 토폴로지보다 더 짧은 전송시간이 측정되었다.

스캐터넷의 형성 시간과 패킷의 전송시간을 구한 결과 노드 수가 10개와 15개인 경우는 링이나 트리 토폴로지가 거의 차이를 느끼지 못한다. 그러나 노드 수가 20개를 넘어서 30개가 될 때까지 링과 트리 토폴로지는 시간의 차이는 점차 늘어 난다. 이러한 결과를 놓고 볼 때 모바일 환경에서는 노드의 추가와 삭제가 더욱 빈번하게 일어난다면 시간의 차이는 더욱 크게 날 것이다. 따라서 스캐터넷의 형성 시 트리 토폴로지보다는 링 토폴로지가 모바일 환경에서 더 적합한 방식임을 알 수 있다.

현재 이동 단말기의 수가 급증함에 따라서 이동 단말기들 간의 네트워크가 크게 작용되게 될 것이다. 이때 사용될 근거리 무선 네트워크 기술은 크게 사용되게 될 것이고 어느 한 기술이 독점적으로 사용

될 수는 없을 것이다. 그렇게 볼 때 가장 이상적인 것은 무선 랜과 블루투스의 조화이다. 하지만 가장 문제가 되는 것은 두 기술이 모두 같은 대역인 ISM 대역의 2.4GHz를 사용하기 때문에 상호간의 간섭이 문제가 발생한다. 앞으로 무선 랜과 블루투스의 호환성에 초점을 맞춘 연구가 필요할 것이다.

참고문헌

- [1] Bluetooth SIG, "Bluetooth Specification Version 1.0B" <http://www.bluetooth.com>
- [2] Charles E. Perkins. Ad-hoc Networking. Addison-Wseley. 2001.
- [3] The Bluetooth Special Interest Group
"<http://www.bluetooth.com>"
- [4] Jenifer Bary and Charles F. Sturman. Bluetooth:connect Without Cable. Prentice Hall, 2001
- [5] Jaap Haartsen. Bluetooth-the universal radio interface for ad-hoc, wireless connectivity. Ericsson Review, (3):110-117, 1998.
- [6] Brent A. Miller and Chatschik Bisdikian. Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communication. Prentice Hall, 2001.
- [7] j. Haartsen, "Bluetooth- the universal radio interface for ad-hoc wiress connectivity," Ericsson Review, no. 3, 1998. <http://www.ericsson.se/review>.
- [8] D. Johnson and D. Maltz, "Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-05.txt>

저자소개



이대봉(Dae-Bong Lee)

2001년 2월 동의대학교 컴퓨터공학과 공학학사
2003년 2월 동의대학교 컴퓨터공학과 공학석사

※관심분야: Bluetooth, ATM-PON



장종욱(Jong-Wook Jang)

1987년 2월 부산대학교 계산통계학과
1995년 2월 부산대학교 컴퓨터공학과 공학박사

1987년~1995년 전자통신연구원
1999년~2000년 Univ. of Missouri at Kansas City, Post Doc. fellowship
1995년. 3월~현재 동의대학교 컴퓨터공학과 교수
※관심분야: Ad-hoc Network, PON