

블루투스 내장 시스템에서의 효율적인 Inquiry procedure

김은호, 김태윤
고려대학교 컴퓨터학과
e-mail:ehkim@netlab.korea.ac.kr

An Efficient Inquiry procedure on Embedded Bluetooth System

Eun-Ho Kim, Tae-Yun Kim
Dept of Computer Science, Korea University

요 약

본 논문에서는 블루투스 디바이스를 찾는 과정인 inquiry 과정과 문제점에 대해 알아본다. 블루투스 inquiry에서의 지연은 사용자가 불편함을 느끼는 물론, 그에 따른 상당한 전력을 소모하게 된다. 이를 개선하기 위하여 각각의 inquiry 마다 얻은 디바이스 정보를 이용함으로써 전체적인 inquiry의 성능 향상을 이루고, 안정적으로 보다 많은 디바이스 정보를 얻을으로써 사용자는 더욱 편리하게 블루투스를 이용할 수 있다.

1. 서론

무선 통신 기술은 규모 면이나 시장 점유 면에서 커다란 성장을 해 오고 있다. 이러한 무선 통신 기술은 기본적으로 음성 전송을 기반으로 하여 발전하여 왔다. 그러나, 최근의 이동 통신 기술은 점차 음성 위주의 전송에서 무선 멀티미디어 데이터 전송으로 점차 바뀌어 가고 있는 추세이다.

무선 데이터 및 멀티미디어의 수요가 증가하면서 기존의 셀룰러 폰뿐만 아니라 laptop PC, PDA와 같은 다양한 휴대용 장치가 널리 퍼지고 있다. 이러한 휴대용 장치는 기존의 infra-structure없이 개개인이 가정이나 사무실에서 혹은 야외에서 개인의 laptop PC, PDA 등을 이용한 데이터 전송에 대한 요구가 빈번하게 발생하고 있다.

이러한 요구를 바탕으로 나타난 기술이 mobile ad-hoc networking 기술이다. mobile ad-hoc networking이란 고정된 구조를 가지지 않고 자체적으로 구성된 무선 이동 네트워크를 말한다. 목적지가 통신 범위 내에 있다면 노드는 방송 메커니즘을 이용해서 통신하게 되며, 다른 목적지로 데이터를 전송할 수 있게 된다. 이러한 mobile ad-hoc

network를 구성하는 기술 중 하나가 블루투스 기술이다.

블루투스[1]는 무선 데이터와 음성 통신을 위해 널리 개방된 명세이다. 이 기술을 사용하는 디바이스들은 여러 가지 목적으로 근거리 라디오 링크를 연결할 수 있다. 블루투스는 초당 1600번의 hopping을 하는 frequency hopping을 기반으로 한다. 블루투스는 2.402GHz와 2.480GHz 사이의 ISM 밴드를 사용한다. 대체적인 데이터 전송률은 1Mb/s이고, 동기식의 음성 채널과 비동기식의 데이터 채널을 동시에 사용할 수 있다.

모든 블루투스 디바이스들은 연결 설정을 시작할 수 있다. 먼저 각 디바이스는 범위 내의 다른 모듈로부터 정보(유일한 48bit 블루투스 어드레스)를 모아야 한다. 이러한 과정을 Inquiry 과정이라고 한다. 그런 다음 디바이스는 Page 과정을 통하여 7개까지의 다른 디바이스들과 연결할 수 있다. 초기 연결을 시작한 디바이스가 마스터가 되고, 연결된 디바이스들은 pico-net을 형성한다.

본 논문에서는 블루투스 inquiry 과정에 대해 초점을 맞춘다. 이 inquiry 과정은 제공되는 매개 변수

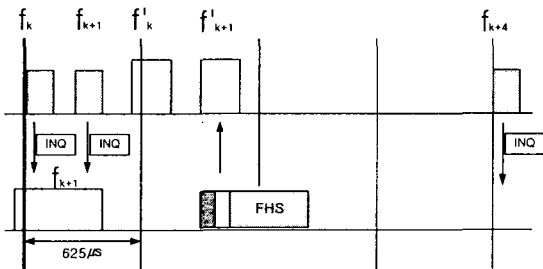
에 따라 디바이스를 찾는 시간이 상당히 차이가 나며, 또한 inquiry 동안 상당한 양의 전력을 소모하게 된다[3,6]. 이러한 디바이스 찾는 시간과 그에 따른 전력 소모는 블루투스 내장 시스템에서는 커다란 장애 요인이 되고 있다.

논문의 구성은 다음과 같다. 2장에서는 블루투스 inquiry 과정에 대해 설명한다. 3장에서는 inquiry 과정의 문제점을 제시하고 이에 대한 해결 방안을 제시한다. 4장에서는 실험 환경 구축과 이를 통한 실험 및 제안한 알고리즘에 대한 성능을 분석한다. 5장에서는 결론을 제시한다.

2. 블루투스 inquiry 과정

블루투스에서의 Inquiry과정은 자신의 주위에 있는 디바이스(프린터, 팩시밀리, LAN의 AP 등)를 찾기 위한 과정이다. 각 디바이스는 연결 설정을 하기 위하여 주위에 있는 다른 디바이스에 대한 정보를 얻어야만 한다. 이러한 Inquiry 과정 동안에 Inquirer는 inquiry에 응답하는 주변 디바이스의 BD_ADDR와 clock에 대한 정보를 얻는다.

inquiry 동안 디바이스는 디바이스 자체 clock과 BD_ADDR에 의해서 얻어진 호핑 순서를 이용한다. 이 호핑 순서는 사용 가능한 79개의 호핑 채널 가운데에서 32개의 채널을 사용한다. 일단 디바이스가 inquiry 호핑 순서를 생성한 후, 호핑 순서에 정의된 각각의 채널에 순차적으로 inquiry 메시지를 broadcast한다. 이 때 inquirer는 inquiry scan state에 들어간다.



<그림 1> 블루투스 Inquiry 과정

위 그림과 같이 inquiry state에 있는 디바이스는 각각의 호핑 rate에 따라 625µs동안 다른 디바이스로부터 응답을 기다린다. inquirer가 ID 패킷을 보내는데 사용되는 32개의 주파수로 구성되는 유일한 호핑 순서가 있다. 이 순서는 각 디바이스마다 유일하

고, 각 phase는 inquiring unit의 native clock CLKN에 의해 결정된다. 더구나 1.28s의 inquiry 호핑 순서는 각각 16개의 주파수로 구성되는 train A, B와 같이 2개의 부분으로 나누어져 있다. inquirer는 한 train 안에 있는 모든 주파수에 대해 패킷을 보내고 가능한 응답을 기다리는데 $T_{train}=10ms$ 의 시간이 걸린다. 블루투스 명세[1]에 의하면 하나의 train은 새로운 train에 들어가기 전에 적어도 $N_{inquiry}=256$ 번을 반복해야 한다.

각각의 블루투스 디바이스들은 주기적으로 inquiry scan substate로 들어간다. 이 substate에서 디바이스들은 역시 자신의 BD_ADDR와 local clock에 따른 호핑 순서에 따라 호핑한다. 이러한 동안 h 호핑 순서 안의 한 주파수에서 inquirer의 ID 패킷을 듣는다. inquiry scan substate에서 inquiry 호핑 순서의 현재 phase는 native clock에 의해 결정된다.

inquiry scan을 하는 디바이스가 ID 패킷을 받으면 inquiry scan substate에서 random backoff time 후에 inquiry response substate로 넘어간다. 이 random backoff time을 두는 이유는 여러 디바이스들이 동시에 같은 주파수에 response 패킷을 보내는 확률을 줄이기 위해서이다. inquiry response substate에서 디바이스는 다시 inquirer의 ID 패킷을 듣는다. inquiry response substate에 있는 디바이스가 주파수 동기를 이룬 후에, 자신의 디바이스 정보(local clock, BD_ADDR)를 포함하는 패킷을 inquirer에게 보내 준다.

3. 블루투스 inquiry의 문제점과 개선 방향

3.1 문제점

블루투스에서는 error-free 환경에서 충분한 응답을 모으려면 세 번의 train 변경이 일어나야 하고, 그 결과로써 inquiry substate는 10.24초의 시간이 걸린다[1]. 또한 error-prone 환경에서는 디바이스를 찾는데 걸리는 최대 시간을 알아내기는 어렵다. 그러나 단지 디바이스를 찾는데 10.24초란 시간은 사용자에게 불편을 줄만큼 상당히 긴 시간이다.

또한 디바이스를 찾는데 오랜 시간이 걸리는 것은 그에 따른 전력 소모가 크다는 것을 의미한다. 또한 [3]에서는 inquiry 과정에 소모되는 전력이 패킷 전송시 소모되는 전력보다 훨씬 크다는 것을 보여준다. 이러한 전력 소모는 결국 블루투스 내장 시스템의 배터리 사용 시간을 단축시킨다.

위의 두 가지 문제점은 블루투스 내장 시스템에

서의 커다란 단점이다[2]. 따라서 블루투스 내장 시스템에서는 최소한의 inquiry 시간 동안에 최대한의 블루투스 디바이스를 찾아야 할 필요가 있다.

3.2 효율적인 inquiry procedure

본 절에서는 보다 효율적인 블루투스 inquiry 과정을 제안한다. 이 알고리즘은 블루투스 베이스 밴드의 변경 없이 HCI 매개 변수의 조작만으로 inquiry 과정의 효율성을 높이는 데 있다.

실제 블루투스는 반경 10m 이내의 근거리 통신을 위한 것이기 때문에 일반적으로 사용자는 자신의 주변에 있는 블루투스 디바이스에 대한 정보를 가지고 있을 것이다. 사용자가 특정한 디바이스(프린터, 팩스, PDA 등)를 찾을 때나, 회의실의 경우와 같이 scatter-net을 구성하려고 할 때 디바이스 inquiry를 시작할 것이다.

제안하는 inquiry 알고리즘에서는 Inquiry_Table이란 구조를 추가한다. 이 Inquiry_Table은 각각의 단계에서 얻은 모든 디바이스의 정보를 가지고 있다. 이 정보를 이용하여 보다 빨리 주변에 있는 모든 디바이스를 찾을 수 있게 된다.

알고리즘은 다음과 같다.

Procedure Inquiry()

```
begin
    Receive Inquiry parameters (Inquiry_Length,
                               Num_Response)

    if new inquiry detected
        Initialize all Inquiry_Table
    end if

    HCI_Inquiry(Inquiry_Length, Num_Response)
end
```

Procedure Inquiry_Complete()

```
begin
    if first response
        Store response to Inquiry_Table
    else
        Check if new device detected
        if new response
            Append new device to Inquiry_Table
        end if
    end if
end
```

```
Inform to UI detected device
end if
end
```

< 그림 2 > 효율적인 inquiry 알고리즘

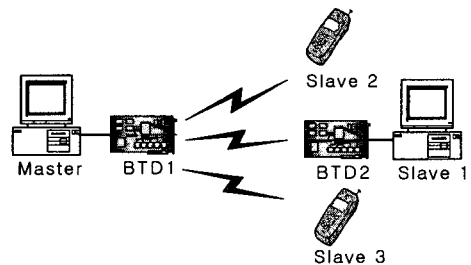
위의 알고리즘에서 사용자는 inquiry 과정을 시작할 때 응답을 기다리는 디바이스(Num_Response)에 대한 정보를 입력할 필요가 있다. 만약 자신의 주위에 있는 디바이스에 대해 정확히 알고 있다면, Num_Response는 총 디바이스의 수가 될 것이고, 모른다면 디폴트 값(초기 사용자 정의 값)으로 정해진다.

다른 하나의 매개 변수인 Inquiry_Length는 많은 고려 사항이 있다. 이 값이 작으면 충분한 디바이스 정보를 얻기도 전에 inquiry 과정이 끝나게 될 것이다. 반대로 이 값이 너무 크다면 불필요한 inquiry 시간이 소비될 것이다. 10개의 디바이스에 대한 inquiry 시뮬레이션 결과[5]와 4개의 디바이스로 실제 환경에서의 실험의 결과 가장 이상적인 값은 5.12초이다. 따라서 특별한 조건인 없는 한 이 값을 기본 값으로 정한다.

각 단계의 inquiry가 끝날 때마다 찾은 모든 디바이스들은 사용자에게 알려준다. 이 정보를 보고 사용자는 계속 inquiry를 할 것인지 연결을 할 것인지 결정하게 된다.

4. 성능 평가 및 분석

2장, 3장에서 설명한 블루투스 inquiry에 대한 성능 평가를 위한 실험 환경은 다음과 같다.

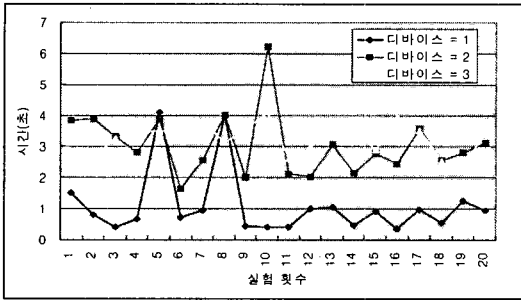


< 그림 3 > 블루투스 inquiry 실험 환경

기본적인 inquiry 및 데이터 전송에 대한 시간은 PC에 RS-232C로 연결된 블루투스 디바이스를 이용하여 측정한다. 즉, PC부분에는 블루투스 프로토콜과 시뮬레이션을 위한 코드가 들어 있고, 블루투스 디바이스에는 블루투스 RF와 베이스 밴드가 들어

있다. 또한 inquiry procedure에 대한 실험을 위하여 2대의 블루투스 내장 셀룰러 폰을 이용한다.

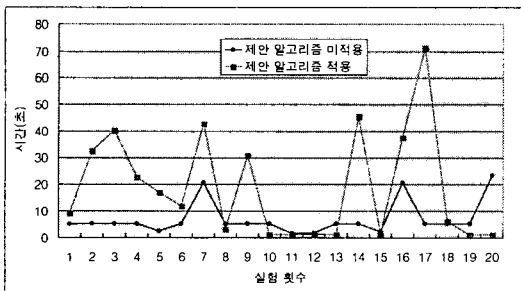
이 실험 환경을 가지고 먼저 Inquiry_Length를 10.24초로 고정시키고 Num_Response를 변화하면서 inquiry 시간을 측정하였다.



< 그림 4 > Inquiry 시간 측정 결과

위의 결과에서 보듯이 찾으려는 디바이스가 작을 수록 inquiry 시간이 줄어드는 것을 볼 수 있다. 특히 찾으려는 디바이스가 1개일 때와 2개, 3개일 때는 inquiry 시간이 많은 차이가 난다. 이 결과는 사용자가 자신 주위의 디바이스에 대한 정보를 알고 있다면 inquiry의 성능을 향상시킬 수 있음을 보여 준다.

또한 제한한 알고리즘의 성능을 평가하기 위하여 다음과 같은 조건을 주고 실험을 하였다 (Num_Response = 3, Inquiry_Length = 2.48). 이 조건은 각 Inquiry_Length(=2.48s)마다 일반적으로 1~2개의 블루투스 디바이스를 찾을 수 있는 시간이다.



< 그림 5 > 제안 알고리즘 적용 결과

위의 결과에서 볼 수 있듯이 제한한 알고리즘은 inquiry 시간이 전체 디바이스를 찾을 수 있는 시간

보다 작게 설정되었더라도 한 번의 inquiry 과정을 거침으로써 원하는 전체 디바이스를 찾을 수 있게 된다. 반면, 기존의 inquiry 알고리즘은 한 번의 inquiry로 모든 디바이스를 찾으려고 함으로써 inquiry 시간에 따라 찾을 수 있는 디바이스의 수가 제한된다. 즉, 이 실험 결과는 열악한 환경에서 제한한 알고리즘은 더욱 뛰어난 성능을 발휘하는 것을 알 수 있다.

위의 실험 결과를 종합해 보면, 사용자는 주위의 블루투스 환경에 대해 이해하고 정확한 디바이스 수를 입력하고, 제안한 inquiry 알고리즘을 적용함으로써 블루투스 inquiry의 성능을 크게 향상시킬 수 있음을 알 수 있다.

5. 결론

본 논문에서는 블루투스 inquiry procedure에 대한 문제점과 이를 개선하는 알고리즘을 제시하였다. 실제로 블루투스 inquiry에서의 지연은 사용자가 불편함을 느끼는 물론, 그에 따른 상당한 전력을 소모하게 된다.

이를 개선하기 위하여 각각의 inquiry 마다 얻은 디바이스 정보를 이용함으로써 전체적인 inquiry의 성능 향상을 이루고, 또한 안정적으로 보다 많은 디바이스 정보를 얻을 수 있었다. 이 정보를 바탕으로 사용자는 더욱 편리하게 블루투스를 이용할 수 있을 것이다.

참고문헌

- [1] Bluetooth Special Interest Group: Specification of the Bluetooth System, Core. Version 1.1 February 2001
- [2] Frank Siegemund and Michael Rohs: Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices
- [3] Oliver Kasten, Marc Langheinrich: First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network
- [4] <http://www.xilinx.com>
- [5] Jan Beutel, Oliver Kasten: A Minimal Bluetooth-Based Computing and Communication Platform