블루투스 HCI 설계 및 성능 분석

류수형, 김 식
세명대학교 대학원 전산정보학과, 세명대학교 정보통신학과

Design and Performance Evaluation for Bluetooth Host Controller Interface

Soo-Hyung Liu*, Shik Kim**
* Dept. of Computer Information Science, Semyung University
** Dept. of Information & Communication Systems, Semyung University
E-mail : shliu@telcom.semyung.ac.kr, shikm@telcom.semyung.ac.kr

요 약

근거리 무선 통신 기술의 하나인 블루투스 기술은 최근 많은 연구와 개발이 이루어지고 있으며, 블루투스 프로토콜 스택의 개발은 블루투스 기술의 응용을 위한 기술적 독립을 의미한다. 이에 대해 본 논문에서는 블루투스 하드웨어와 호스트 사이의 인터페이스 계층으로 블루투스 프로토콜의 HCI 계층을 설계하였고, 일련의 실험을 통하여 기본 동작 과정 및 기 개발된 타사의 블루투스 어뎁터와의 동작 호환성도 연구해 보았다.

1. 서론

블루투스(Bluetooth)는 전자장비간의 케이블을 없앨 목적으로 스웨덴의 에릭슨(Ericsson)사에서 연구되기 시작한 근거리 무선 통신 기술의 하나이다. 최근에는 블루투스 프로토콜 스택의 표준규격을 책정하기 위한 목적으로 1998년 에릭슨, 노키아(Nokia), IBM, 도시바(Toshiba), 인텔(Intel) 5개 사를 주축으로 Bluetooth SIG(Special Interest Group)라는 컨소시엄을 구성했으며, 현재는 회원사가 전세계적으로 2000여 개에 이를 만큼 차세대 무선 통신 기술로서 블루투스에 대한 업계의 관심이 높아지고 있다[1-4].

블루투스 기술을 응용한 제품을 개발하기 위해 참여하는 기업들이 급속도로 증가하고 있으나, 블루투스의 핵심기술에 대한 국내기술수준은 초보 단계에 있다. 따라서 블루투스 프로토콜 스택을 개발하는 것은 소프트웨어 응용단계에서 핵심기술에 대한 부분적인 기술 독립을 의미하는 것이다. 블루투스 프로토콜 스택은 하드웨어와 소프트웨어로 구성되는 코어(Core) 부분과, 블루투스 기술의 각종 응용을 위한 프로파일로 나뉘어 진다. 본 논문은 블루투스 프로토콜 스택을 분석하여 블루투스 기술의 여러 가지 응용을 위한 첫 단계 과제로 하드웨어와 소프트웨어의 인터페이스 역할을 하는 HCI(Host Controller Interface) 계층을 설계하고, 단위 기능 테스트를 위한 프로그램을 작성하여 HCI 계층의 기본 동작실험을 하고, 설계된 HCI가 타사의 블루투스 장치와도 호환 가능한지를 연구하고자 한다.

본 연구는 (주)진두네트워크의 연구과제 일환으로 수행 되었음

2. 블루투스 HCI 분석

블루투스 프로토콜 스택의 구조를 통해서 HCI 계층이 상·하 위 프로토콜들과 어떤 관계에 있는 지와 설계에 필요한 구조에 대해서 논의한다.

그림1은 일반적인 블루투스 프로토콜 스택의 기본 구조 및 HCI 계층의 구조를 나타내고 있다. 최하위 계층인 Radio부터 Baseband, LMP(Link Manager Protocol)계층과 HCI 계층의 일부는 블루투스 시스템의 하드웨어 및 펌웨어를 구성하는 모듈로서 장치의 특성을 정의하고, 물리적 무선 연결과 생성을 관리하며, 불필요한 신호들에 대한 필터링기능 등을 제공한다. 블루투스 프로토콜 스택에서 하드웨어 계층과 소프트웨어 계층으로 양분되어 있는 HCI 계층은 블루

투스장치와 호스트 사이의 인터페이스 역할을 하는 계층으로 전송되는 패킷들의 타입을 정의하고 있다.

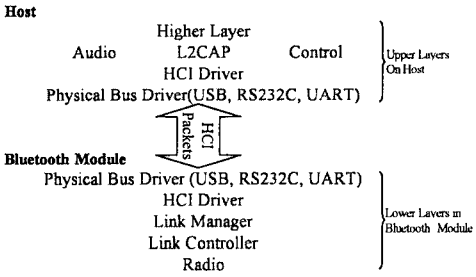

그림1. 블루투스 프로토콜 스택내의 HCI 구조

L2CAP(Logical Link Control Adaptation Layer)은 프로토콜 종류를 구분하기 위한 멀티플렉싱과 송·수신 데이터에 대한 패킷분할 및 재조합 그리고 QoS(Quality Of Service)정보의 전달을 목적으로 한다. SDP(Service Discovery Protocol)는 블루투스 장치들 상호간에 제공하는 서비스와 속성을 알아내는 역할을 하며, Audio는 BaseBand계층을 통해 직접 음성데이터를 송·수신한다. 그 밖의 상위 프로토콜 계층들은 블루투스 장치의 응용과 관련되는 프로토콜들이다[5-6].

2.1. HCI 계층의 구조와 동작

그림1과 같이 HCI 계층은 블루투스 하드웨어 부분과 호스트의 소프트웨어 부분으로 나뉘어져있다. 따라서, 블루투스 하드웨어는 물리적인 버스(USB, RS232C, UART 등)를 통하여 호스트 시스템과 패킷 교환을 수행하게 된다. 교환되는 패킷은 Command 패킷, 송신 데이터 패킷, 수신 데이터 패킷, Event 패킷의 네 가지 이다. 이 때 호스트 시스템으로 전달되는 Event와 수신 데이터 패킷은 비동기 적으로 전달되며, 특히 데이터 전달은 표준규약에 따른 일련의 흐름제어 메커니즘을 필요로 한다[3-5]. 다음은 흐름제어 및 각 패킷의 구성과 설계를 위한 고려 사항이다.

- Command 패킷은 총 6종류 95가지로 구성 되어있으며, 구현을 위해서는 모든 Command를 HCI의 표준규약에 정해진 형태로 구성하여 버스 드라이버로 전달할 수 있는 기능이 필요하다.
- Event 패킷은 총 32가지이며, 구현을 위해서는 버스 드라이버로부터 비동기 적으로 전달되는 모든 Event 패킷을 받은 후, 프로토콜 스택의 상위계층으로 전달하는 기능이 필요하며, 이때 Event 패킷은 반드시 해석(Decode)된 것이어야 한다.
- 송·수신 데이터 패킷에 대한 구현에는 전송할 데이터를 HCI 데이터 패킷의 표준규약 형태로 구성한 후, 버스 드라이버로 전달 할 수 있어야 하고, 버스 드라이버로부터 비동기 적으로 전달되는 모든 HCI 데이터 패킷을 받은 후, 프로토콜 스택의 상위계층으로 전달하는 기능이 필요하다. 이때, 전달받은 HCI 데이터 패킷들은 상위 계층의 요구에 맞게 재구성되어야한다.
- 표준규약에 따른 흐름제어는 처음 데이터를 전송할 때 호스트 측에서 최초에 Read_Buffer_Size Command를 이용하여 전송버퍼의 크기를 구한 후, 버퍼크기의 범위 내에서 데이터 패킷을 전송하게 된다. 이후 전달된 패킷에 의해서 Number_Of_Completed_Packet Eevent가 발생하고, 이 Event정보를 통해 추가로 전달할 수 있는 패킷의 개수를 갱신하게 된다. 따라서 흐름제어의 구현을 위해서는 데이터 패킷을 보내는 스레드와 이에 상응하는 Event를 받는 스레드를 독립적으로 구현해야 하며, 이들 사이의 정보전달을 위한 대책이 마련되어야 한다.

3. HCI의 설계

HCI의 설계는 객체지향설계가 갖는 장점을 갖추기 위해 C++언어에서 제공하는 객체지향구조를 기반으로 했다. 먼저 블루투스 표준규격에서 제공하는 기능을 기능성을 기반으로 분할하여 HCI 클래스구조를 설계하였다. 설계된 HCI의 클래스 구조는 HCI_Command 클래스, HCI_Event 클래스, Pipe 클래스, HCI_Stack 클래스로 구성된다. 다음은 설계를 위한 기능 분석 및 설계된 각 클래스에 대한 정의와 Command전송에 필요한 동작 메커니즘 그리고 데이터 송·수신을 위한 흐름제어에 대해 논의한다.

3.1. HCI 계층의 기능성 분석

블루투스 프로토콜 스택에서 HCI계층이 담당해야 하는 기능을 상호관계와 흐름제어를 기반으로 분할하면 그림2와 같다. 그림2에 표현된 HCI의 기능은 다음과 같다[7].

- 상위계층에서 요구하는 서비스는 HCI계층에서 일련의 HCI명령들로 변환한 후 HCI명령패킷들을 구성하여 버스드라이버로 전달한다.

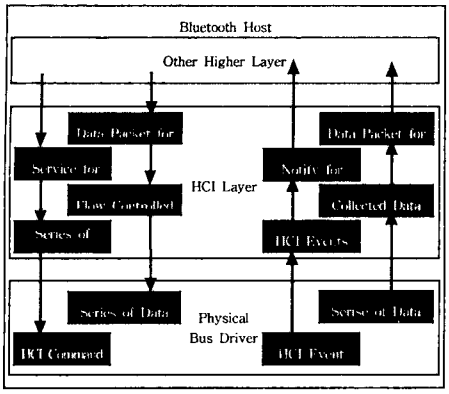

그림2. HCI 계층의 기능 및 동작 메커니즘

- 상위계층에서부터 전달되는 데이터패킷은 HCI계층에서, 흐름제어에 대한 처리 하에, 일련의 HCI 데이터패킷들로 구성하여 버스드라이버로 전달한다.
- 버스드라이버에서부터 전달되는 HCI이벤트패킷은 HCI계층에서 분석하여, 상위계층에게 전달해야할 내용들만 통지한다.
- 버스드라이버에서부터 전달되는 데이터패킷은 HCI계층에서 축적하여 상위 계층에서 사용될 수 있는 형태의 데이터패킷으로 재구성하여 전달한다.

분할된 구조를 클래스 구조로 설계할 때 블루투스 플랫폼과 개발환경의 방법론 및 객체지향언어 특성상 고려되어야 요소는 다음과 같다.

- 멀티쓰레드: 버스드라이버로부터 전달되어오는 HCI 이벤트패킷, 수신데이터패킷은 버스드라이버로 전달하는 HCI명령패킷, 송신데이터패킷과 밀접한 관계가 있지만, 완전히 비동기적으로 전달된다. 따라서, HCI명령패킷과, 송신데이터패킷을 처리하는 프로세서와, HCI이벤트패킷, 수신데이터패킷을 처리하는 프로세서는 별도의 쓰레드로 구현되어야 한다. 이러한 처리를 위해서는 멀티쓰레드를 이용하는 프로그래밍 기법이 필요하다.
- 파이프: HCI계층에서 담당해야 할 흐름제어(Flow Control)기능의 구현을 위해, 데이터패킷을 전달하는 프로세스와 전송된 패킷에 대한 결과신호인 "Number Of Completed Packet Event"라는 이벤트를 받아서 적절한 처리를 하는 프로세스가 독립적인 쓰레드로 구현되어야 할뿐만 아니라, 이들 두 쓰레드간의 체계적인 통신 메커니즘이 필요하다[8].
- 다형성(Polymorphism): HCI계층의 구현에서는 상위 계층의 요구에 따라 95가지의 HCI명령에 대한 명령패킷을 구성하는 기능과, 이벤트패킷으로부터 32가지의 이벤트 중 하나로 분석(Decode)하는 기능이 필수적으로 포함된다. 이 기능에 대한 구현은 다양한 형태로 나타날 수 있으나, HCI명령은 AssembleCommand(), HCI이벤트는 MakeEvent()와 같이 특정의 공통적인 인터페이스로 구현된다면 수많은 HCI명령과 HCI이벤트들을 체계적으로 관리와 확장성을 유지할 수 있다.
- CPU 스케줄: HCI계층의 흐름제어를 위해 구현된 두 쓰레드는 간접적으로 상대방의 트리거(Trigger)를 기다리고 있는 구조이다. 즉, 데이터패킷을 전달하는 쓰레드는 "Number Of Completed Packet Event"라는 이벤트가 처리되어 패킷을 추가로 보낼 수 있는 상태를 기다리고 있고, "Number Of Completed Packet Event"라는 이벤트는 데이터패킷이 전달된 후에 발생하는 것이다. 만약, 이벤트를 처리하는 쓰레드가 "Number Of Completed Packet Event"라는 이벤트를 전달받아 처리할 정도의 CPU의 서비스를 받지 못한다면, 쓰레드들은 교착상태에 빠지게 된다. 흐름제어를 위하여 동작할 코드는 CPU의 서비스를 고르게 받을 수 있도록 단위 쓰레드를 설계해야 한다[9].

3.2. HCI의 계층구조 설계

블루투스 표준규격에서 제공하는 기능을 기능성을 기반으로 분할하여 HCI 클래스구조를 설계하였다. 그림3과 같이 설계된 HCI의 클래스 구조는 다음과 같다.

- HCI_Command 클래스: Command의 일반적인 형식을 정의한 클래스이며, 하위 클래스로 HCI에 규정된 95개의 Command클래스 정의와 멤버함수로 AssembleCommand함수를 등록하여 각 Command 패킷들의 인스턴스를 구현할 수 있도록 한다.
- HCI_Event 클래스: 블루투스 장치로부터 발생되는 Event코드의 일반적인 형식을 정의한 클래스이다. 하위 클래스로 HCI에 규정된 32개의 Event클래스 정의와 멤버함수로 MakeEvent함수를 등록하여 각 Event패킷들의 인스턴스를 구현할 수 있도록 한다.
- Pipe 클래스: 호스트와 블루투스 장치간의 Command 전송과 데이터 송·수신을 위한 쓰레드의 유기적인 동작과 흐름제어를 위해 사용되는 IPC(Inter-Process Communication) 메커니즘 중의 하나인 파이프(Pipe)를 구현한 클래스이다.
- HCI_Stack 클래스: 위에서 정의된 클래스들을 기반으로 하여, Command실행에 따라 발생한 다양한

Event에 대해서 콜백함수를 등록하여 호스트가 Event응답을 받을 수 있게 했고, 흐름제어를 통해서 데이터 패킷을 송·수신하도록 구현할 수 있도록 한다.

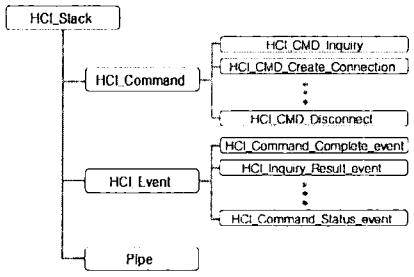

그림3. HCI의 클래스 계층구조

3.3. Event기반의 쓰레드 동작 메커니즘

블루투스 프로토콜 표준규약에 따르면 Command의 실행에 대한 결과로 다양한 Event가 비동기 적으로 발생한다는 것을 알 수 있다. 따라서, 발생되는 Event들에 대한 비동기 적인 처리메커니즘이 필수적이다. 이 논문에서는 비동기 적으로 발생되는 다양한 Event들을 처리하기 위해 그림4와 같은 Event기반의 스레드 동작 메커니즘을 제안하였다[3-4].

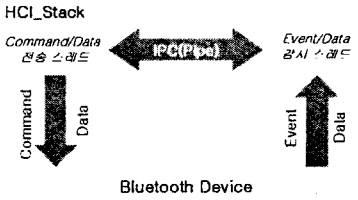

그림4. Event기반의 스레드 동작 메커니즘

이 메커니즘은 다중 스레드 환경에서 발생할 수 있는 스레드 동기화에 관련된 문제를 해결하기 위한 것으로, Event를 감시하는 스레드는 블루투스 장치로부터 Event가 발생했을 경우 이에 대한 정보를 파이프 객체를 이용해서 Command를 전송하는 스레드에게 다음 Command를 실행 할 수 있도록 통지하는 역할을 한다. 또한 호스트에서 블루투스 장치로 전달하는 데이터 패킷은 블루투스 장치의 데이터 버퍼가 넘치지 않도록 보내야 한다. 이러한 일련의 과정을 흐름제어라 하며, 본 논문의 HCI 흐름제어 부분은 다음과 같이 설계하였다.

그림4와 같이 데이터 패킷을 블루투스 장치로 전달하는 스레드와 Number_Of_Completed_Packet Event를 전달받는 스레드를 각각 독립적으로 동작하도록 구성하였으며, 이들 간의 정보전달을 위하여 파이프 객체를 사용하였다.

4. HCI 성능 분석

블루투스 HCI의 실험을 위해 사용된 장치는 MMC Technology 사의 블루투스 어댑터와 Ericsson사 및 Widcomm사의 블루투스 어댑터를 혼용하여 통신을 수행하였고, USB드라이버와 통신하기 위해 CSR (Cambridge Silicon Radio)에서 제공되는 BlueCore01 USB Stack 드라이버 모듈을 사용했다. 다음에 나오는 그림5, 그림6과 그림7은 HCI의 동작을 검증하기 위해 설계된 HCI 클래스 구조를 사용하여 개발된 HCI_Test 프로그램으로 실험한 일련의 결과들을 보여 주고 있다.

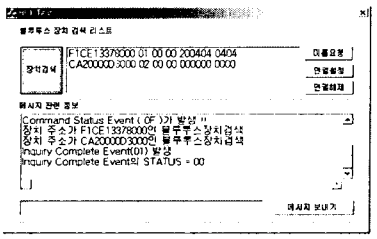

그림5. 블루투스 장치 검색 실행화면

HCI_Test 프로그램을 실행하여 블루투스 장치를 검색한 화면은 그림5와 같다. 장치검색 버튼을 누르면 일련의 Command가 블루투스 장치로 전송되며, 상단 리스트 박스에는 이에 대한 결과로 검색된 장치들의 주소가 출력되고, 중간의 메시지 박스에는 이에 대한 동작과정으로 블루투스 장치로부터 발생한 Event코드가 표시된다. 이러한 결과로 보아 본 논문에서 설계한 Command 및 Event를 처리하는 메커니즘이 제대로 수행되고 있음을 확인 할 수 있다.

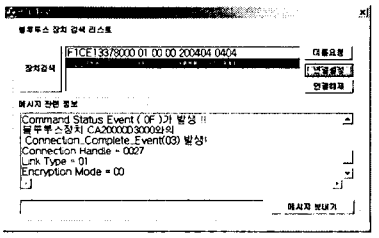

그림6. 블루투스 장치간 연결설정 실행화면

그림6은 검색된 블루투스 장치리스트 중 원하는 블루투스 장치와 연결을 설정한 화면이다. 리스트 박스에서 장치를 선택하고 연결설정 버튼을 누르면 선택된 블루투스 장치와의 연결이 이루어지고 연결된 장치의 주소와 발생한 Event코드 정보가 메시지 박스에 성공적으로 표시되었다.

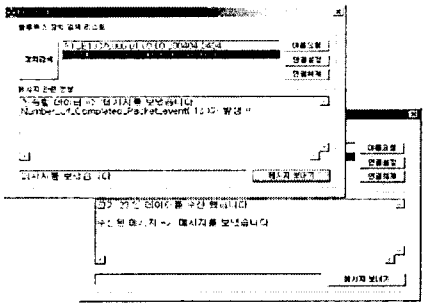

그림7. 블루투스 장치간 데이터전송 실행화면

그림7은 연결이 설정된 장치들 사이에서 데이터를 전송한 화면이다. 연결 설정 후 메시지를 입력하고 메시지 보내기 버튼을 누르면, 연결하고 있는 다른 블루투스 장치로 데이터가 전송된다. 수신측 호스트는 보내온 데이터 패킷의 순서와 크기에 맞게 패킷을 받게 되고, 전송 측 호스트는 전송이 확인된 데이터 패킷마다 성공적인 Event가 발생한다. 실험 데이터에는 나타나 있지 않지만 실제 64Kbytes 이상의 큰 데이터를 무한 루프로 작동시킨 결과 두 장치간의 데이터 전송이 정상적으로 이루어 졌으며, 이것으로 두 장치간의 데이터통신에 흐름제어가 이루어지는 것을 확인 할 수 있었다.

5. 결론

블루투스는 차세대 근거리 무선통신 기술로 각광받고 있으며, 이것은 프로토콜 스택의 개발을 통한 다양한 서비스제공을 필요로 하게 된다. 블루투스 표준규격은 무선분야부터 응용분야까지 전 시스템이 공개구조를 채택하고 있다. 그러나, 블루투스 프로토콜 스택은 개발자에 따라 다른 방법으로 하드웨어와 소프트웨어의 역할을 기능성을 중심으로 독자적으로 분할하여 설계하도록 한다. 본 연구는 블루투스 프로토콜을 획득하기 위한 첫 단계 연구과제로서 상위계층 프로토콜 스택 개발을 위한 HCI를 설계하였고, 표준규약에 따르는 HCI의 동작 과정을 검증하기 위한 일련의 실험들을 수행하였다. HCI를 설계하기 위하여 블루투스 표준규약을 분석하여 HCI 클래스 구조를 설계하였고, 설계된 HCI 클래스 구조를 블루투스 운영플랫폼 상에서 설계하기 위한 개발 환경의 고려 사항을 확인하였다. 설계된 HCI 클래스 구조를 사용하여 일련의 실험을 수행하였다. 그 결과 HCI 계층의 기본 동작과정으로 블루투스 장치검색, 연결설정 및 데이터 전송이 모두 성공적으로 수행되는 것을 확인하였다. 또한 기 개발된 Ericsson사의 어뎁터와 WIDCOM사의 어뎁터 등과의 통신 실험을 통하여 본 논문에서 설계된 HCI는 구조의 범용성과 호환성이 가능하다는 것이 검증되었다.

[참고문헌]

[1] Zenocom, Zenocom Document Website, <http://www.zenocom.co.kr>

[2] 3Com, 3Com Document Website, <http://www.3com.co.kr/mobile/wireless/>

[3] Bluetooth SIG, The Bluetooth Specification v1.1, <http://www.bluetooth.com>

[4] CSR, BlueCore01 USBStack Document <http://www.csr.com>

[5] Jennifer Bray and Charles F. Sturman, "Bluetooth Connect Without Cables", Prentice Hall, 2001

[6] James Y. Wilson and Jason A. Kronz, "Inside Bluetooth Part I", Dr. Dobb's Journal, 2000

[7] Jaap C, Haartsen, "BLUETOOTH™ : A new radio interface providing ubiquitous connectivity", IEEE VTC2000, p107-111, 2000

[8] RON SCHNEIDERMAN, "Bluetoth's slow dawn", IEEE SPECTRUM November, pp 61-65, 2000

[9] Zhang Pei, "Bluetooth - The Fastest Developing Wireless Technology", IEEE VTC2000, pp 1657-1664, 2000