

Wi-Fi Direct 기반 무선 Docking 시스템을 위한 Bluetooth Tunneling 연구

이 재 호*

Bluetooth Tunneling Method for Wireless Docking System Based on Wi-Fi Direct

Jaeho Lee*

요 약

무선 Docking 시스템은 기존에 설치된 모니터, 키보드 등의 별도 환경을 통하여 스마트폰 등의 이동형 컴퓨팅 장치 사용에 편의성을 제공하는 환경으로써, 사용자는 Docking 시스템이 구축된 어느 환경에서라도 자신의 이동형 컴퓨팅 장치를 다양한 주변 장치들을 활용하여 편리하게 사용할 수 있다. Docking 시스템은, 컴퓨팅 장치를 중심으로 구축된 기존 데스크탑 및 Laptop 환경과 달리, 컴퓨팅 장치를 제외한 주변 장치들로 구성되어 있다. Wi-Fi Alliance에서는 이러한 환경을 제공하기 위하여 Wi-Fi Docking 표준화를 최근 진행 중이며, 기존에 발표된 Wi-Fi Direct 표준 기술(Wi-Fi Peer-to-Peer Technical Specifications v1.2, 2010)을 활용하고 있다. 하지만, Docking 시스템에서 Bluetooth 장치를 사용할 경우 인터페이스 상이로 인하여 복잡한 Connectivity가 발생할 수 있다. 본 고에서는 이러한 문제를 해결하기 위하여 Wi-Fi Direct 환경에서 Bluetooth 장치 연결을 Tunneling 할 수 있는 기술을 설계하고 실험을 통하여 분석하였다.

Key Words : Wi-Fi Direct, Bluetooth, Docking, Tunneling, Wireless Serial Bus.

ABSTRACT

Wireless Docking system can provide enhanced convenience to user experience of handheld device such as smart phone by using previously deployed peripheral devices such as monitor and keyboard. In this environment, user can easily use the handheld device with variable peripheral devices at any docking system place. This system would be composed of peripherals except host computing device contrarily to previous desktop and laptop environment. For this system, Wi-Fi Alliance has been developing standard technology based on Wi-Fi Direct(Wi-Fi Peer-to-Peer Technical Specifications v1.2, 2010) technology. However, this system can make a problem which may lead to complex connectivity on handheld device due to non-compatible communication interface. To address given problem, we designed a new method of Bluetooth tunneling technology via previous Wi-Fi Direct communication, and evaluated it with experiment results.

I. 서 론

Docking 시스템은 스마트폰 보급과 함께 가정용

하이파이 스피커 등의 고사양 인프라 장비를 활용하여, 스마트폰 음악 감상이나 핸드프리, 알람 등의 기능을 활용하여 사용자 환경을 개선하거나 편의성 증

* First Author : Seowon University, Department of Information and Communications Engineering, izeho@seowon.ac.kr, 정희원
논문번호 : KICS2016-08-188, Received August 5, 2016; Revised October 18, 2016; Accepted November 9, 2016

진을 위하여 사용되는 기본적인 스마트폰 시스템이다. 이 환경을 통하여 사용자는 고음질과 고출력의 음악을 감상할 수 있고, 휴대폰을 직접 소지하지 않아도 편리하게 통화할 수 있으며, 대표적으로 Docking 기능이 내장된 스피커 제품 등이 현재까지 출시되어왔다. 하지만, 이러한 Docking 시스템은 스마트폰 등의 개인 휴대장치의 음악 또는 영상을 외부 스피커와의 USB 연결 또는 Docking Station에 직접 장착 등을 통하여 물리적으로 연결해야 하는 환경적 제약이 따른다.

이러한 제약을 해결하기 위하여 최근 Wi-Fi Alliance 에서는 Wi-Fi 인터페이스를 사용하여 무선 Docking 시스템을 위한 Wi-Fi Docking^[1] 표준화를 진행하고 있다. 이 기술은 기존에 발표된 Wi-Fi Direct^[2] 표준 기술을 활용하여 Docking 시스템 기능을 개발하고 있으며, IEEE 802.11ad^[3] 기반의 WiGig^[4] 표준에도 적용될 예정이다.

Wi-Fi Direct 표준 기술은 홈네트워크 및 근거리 네트워크 환경에서 AP(Access Point) 등 별도의 통신 인프라 장비 없이도 장치간 직접 연결을 통하여 데이터 송수신을 지원함으로써, 최근 Wi-Fi Display^[5] 등의 응용 기술로 활발히 적용되고 있다. 이 기술을 활용하면 AP가 존재하지 않는 환경에서 무선 Docking 기능을 용이하게 구현할 수 있으며, 특히 Docking 시스템의 주변 장치를 구성하는 Docking Cluster를 대표적으로 제어하는 Docking Station과 스마트폰 등의 사용자 이동형 컴퓨팅 장치를 손쉽게 연결할 수 있는 장점을 제공할 수 있다.

현재 개발 중인 Wi-Fi Docking 표준에 의하면, Docking 시스템은 모니터, 키보드, 프린터 등의 주변 장치를 나타내는 Peripheral 장치, 다수의 Docking Peripheral 장치와 사용자 컴퓨팅 장치간 통합 연결성을 지원하는 Docking Center 장치, 그리고 사용자 컴퓨팅 장치를 의미하는 Dockee 장치로 구분된다. 이때 Peripheral 장치들은 대부분 USB 인터페이스로 구성되어 있는 경우가 보편적이기 때문에, Wi-Fi Docking 표준은 USB 인터페이스를 Wi-Fi를 통하여 무선으로 전달할 수 있는 Wi-Fi Serial Bus(WSB)^[6] 기술을 Wi-Fi Direct 기반으로 활용하는 방향으로 표준 기술을 개발 중이다.

하지만, 최근 Bluetooth^[7] 기술을 사용하는 키보드, 마우스 등의 HID(Human Interface Device) 장치들이 매우 높은 시장점유율과 함께 성장하고 있으며 그 활용도 또한 매우 높다. 또한 Bluetooth SIG(Special Interest Group)에서 지난 2010년에 발표한 Bluetooth LE(Low Energy)^[8] 기술이 높은 관심을 받고 있으며,

현재 비콘 등의 저전력 근거리 통신 시장 점유율을 급격하게 높이고 있다. 이 기술은 최근 매우 깊은 관심을 받고 있는 IoT (Internet of Things) 테마와 함께, 홈 네트워크 환경의 대표적인 통신 기술로 자리잡고 있으며, 향후 HID 외에도 Docking 시스템을 구성하는 각각의 Peripheral 장치로 활용될 가능성이 매우 크다. 또한 Bluetooth LE기술은 트래픽 패턴에 따른 센서네트워크 라우팅^[10] 기술에도 적용이 용이하며, Wi-Fi Direct의 장치간 직접 통신^[11] 환경에도 저전력 통신에 특화된 통신 방안^[12]을 구현하기에 적합하다. 이러한 관점에서, Wi-Fi Docking 기술은 현존하는 USB 인터페이스 외에도 Bluetooth 및 Bluetooth LE 기술을 지원할 필요성이 매우 높지만, Bluetooth 등의 이기종 무선 통신을 그대로 사용하기에는 많은 환경적 한계가 있다.

Docking 시스템을 구성하는 Dockee 장치 관점에서, 기존의 USB를 지원하는 Peripheral 장치의 경우 현재 개발 중인 표준 기술을 그대로 활용하여 Docking Center로 통합 연결할 수 있다. 하지만, Bluetooth 및 Bluetooth LE를 지원하는 Peripheral 장치의 경우 Dockee 장치가 해당 Peripheral 장치와 직접 Bluetooth 및 Bluetooth LE 연결을 수행해야 하는 오버헤드가 발생되고, 이는 향후 출시될 Bluetooth LE를 지원하는 다양한 IoT 제품들을 고려할 때 Connectivity 관점에서 매우 높은 복잡성을 야기시킬 수 있다.

본 논문에서는 이러한 문제를 해결하기 위하여, Wi-Fi Docking Center에서 Bluetooth 및 Bluetooth LE 인터페이스를 그대로 지원할 수 있도록 Bluetooth Tunneling 기법을 제안한다. 본 고에서 제안하는 기법은, Bluetooth Host 와 Bluetooth Controller 간 통신 규격인 HCI 정보를 그대로 원격 장치에 Tunneling 하는 방안을 제시함으로써 Wi-Fi Serial Bus 및 Bluetooth 기술 모두에 호환될 수 있으며, 또한 이기종 간 통신 기술 융합 측면에서 명료한 방안을 도출함으로써 구현 상의 단순함을 제공할 수 있다.

II. Wi-Fi Docking 표준 현황

Wi-Fi Docking 기술은 스마트폰 등과 같은 사용자 휴대 장치 사용 환경을 개선하기 위하여, 모니터, 프린터, 마우스, 키보드 등의 고정형 주변 장치를 대상으로 Docking 시스템을 구성하고, 이를 통하여 사용자 휴대 장치에 마치 데스크탑 사용 환경과 같은 편의성을 제공하기 위하여 Wi-Fi Alliance 표준 조직의

Wi-Fi Docking TTG (Technical Task Group)에서 개발 중인 표준 기술이다. 이 표준 기술은 초기 Wi-Fi Docking MTG (Marketing Task Group)에서 2013년 발표한 Wi-Fi Docking MRD(Marketing Requirement Document)^[9] 문서에서 최초로 사용자 Use-Case를 정의함으로써 시작되었고, 같은 해 Wi-Fi TTG가 결성됨으로써 본격적으로 표준기술 개발이 진행되었다.

Wi-Fi Docking MRD 문서에는 Home Office, Enterprise Office, Public Access Workplace, Wireless Meeting Room, Car Docking System, A/V Entertainment, Multiple Dockees 로 구성되는 총 7가지의 사용자 Use-Case가 정의되었고 이를 지원하기 위한 기술적 요구사항을 생성하였다. 또한 위 문서에는 Wi-Fi Docking 시스템을 구성하는 기능 역할을 Dockee, Docking 시스템, Docking Peripheral의 3가지 형태로 구분하였으며, 각 기능 역할에 따른 지원 기술의 종류도 정의하였다.

우선 그림 1과 같은 Home Office 환경에서는, 사용자 휴대장치가 가정 또는 사무실 환경에 설치된 모니터와 키보드 등의 주변장치와 Wi-Fi로 연결되어 사용하는 Use-Case로써, Wi-Fi Docking MRD에서 정의하는 첫 번째 시나리오이다. 이 환경에서의 모니터는 키보드나 마우스, 스피커, 프린터 등과 USB 인터페이스를 통하여 물리적으로 연결되어 있으며, 사용자 휴대장치는 오직 모니터와의 Wi-Fi 연결을 통하여 모니터를 포함한 모든 주변 장치를 사용할 수 있어야 한다.

본 환경에서는, Wi-Fi Docking 시스템을 위하여 기존 주변 장치들이 모두 사용자 휴대장치와의 연결을 위하여 Wi-Fi 인터페이스를 보유할 필요는 없으며, 기존 USB 인터페이스 환경을 모니터 장치가 제공한다는 점을 가정하고 있다. 즉, 모니터 장치는 Wi-Fi Docking 시스템에서 Docking Center 역할을 수행하고, 사용자 휴대장치인 Dockee와의 Wi-Fi 인터페이스를 통하여 기존 주변 장치가 사용하는 USB 인터페이스를 제공한다.

따라서 Wi-Fi Docking 시스템을 이용하는 사용자는 자신의 휴대장치인 Dockee를 Docking Center와



Fig. 1. Home/Office Use-Case of Wi-Fi Docking System[9].

무선으로 연결한 후, Docking Center를 중심으로 설치된 기존 주변장치들을 그대로 사용한다. 이러한 Docking 시스템이 가정과 사무실에 모두 존재한다는 가정 하에, 사용자는 자신의 휴대장치만 보유하면 가정과 사무실에서 동일한 컴퓨터를 사용하는 효과를 얻을 수 있다.

그림 2는 Wi-Fi Docking MRD에서 정의하는 3번째 Use-Case로써, 공항이나 은행, 도서관 등의 공공장소에서 Wi-Fi Docking 시스템을 활용하여 사용자 단말장치를 데스크탑 환경과 동일하게 사용하는 시나리오를 나타낸다. 이 환경에서도 Dockee, Docking Center 및 Docking Peripheral 역할은 동일하게 적용되지만, 사용자 보안문제와 과금 등의 기능이 원활하게 해결되어야 할 것을 요구하고 있다.

이러한 두 가지 Wi-Fi Docking 환경에서는 사용자 휴대장치와 모니터, 주변장치 등 물리적 장치 관점에서의 Docking 역할이 명확하지만, USB 및 스크린 미러링 기능을 수행하는 Wi-Fi Display 기술을 사용할 경우의 각 장치별 역할 정의가 필요하며, 이에 대한 예시를 표 1에 나타내었다.

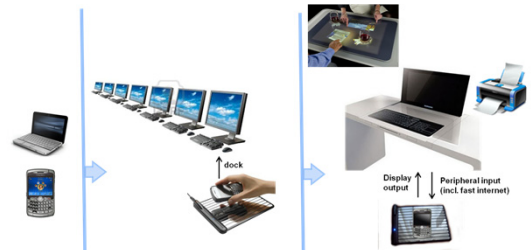


Fig. 2. Public Access Workplace Use-Case of Wi-Fi Docking System[9].

Table 1. Device Role Position for Each Interface Support.

Device Type	Handheld	Monitor	Keyboard, etc.
Docking Role	Dockee	Docking Center	Peripheral
USB Role	WSB host	WSB hub	WSB Peripheral
Wi-Fi Display Role	Display Source	Display Sink	X

III. 제안방식 개요

3.1 시스템 구성 요소

그림 3은 Wi-Fi Docking 시스템의 장치 구성을 나타내었으며, 그림 4는 이에 대한 시스템 구성도를 나타내었다.

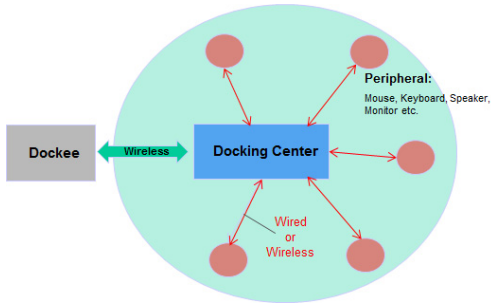


Fig. 3. Device Roles and Components of Wi-Fi Docking System.

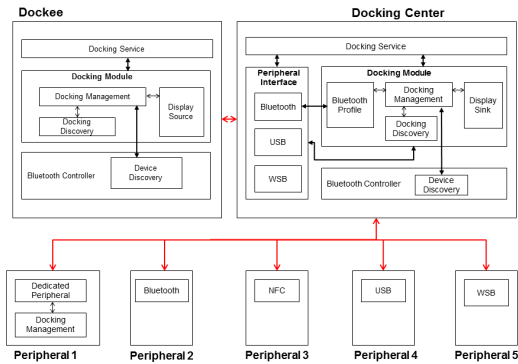


Fig. 4. Wi-Fi Docking System Architecture.

타내었다. Wi-Fi Docking 시스템은 각 장치의 역할에 따라 Dockee, Docking Center, Docking Peripheral 장치로 구분되며, Docking Peripheral 장치는 일반적으로 H/W Resource가 충분한 Docking Center에 연결된다. Docking Center 장치는 일반적으로 모니터 등의 디스플레이 장치에서 수행되고, USB 인터페이스를 통하여 Docking Peripheral 장치와 연결되며 Wi-Fi 인터페이스를 통하여 Dockee 장치와 연결된다. Dockee 장치는 사용자가 활용할 컴퓨팅 장치에서 수행되고, 일반적으로 스마트폰이나 태블릿 PC 등이

Table 2. Device Role Position for Each Interface Support.

Docking Role	Physical Device
Dockee	Smart Phone, Laptop, Tablet PC, Portable Music Player, Portable Game Console, Camera
Docking Center	Monitor, TV, Tablet PC, AP, Router, Gateway
Docking Peripheral	Mouse, Keyboard, Game Pad, Speaker, Microphone, Projector, Portable Display, Car Equipment, Printer, Scanner, Lighting, Home Automation Device, Home Appliance

Dockee 역할을 담당한다. 표 2는 Wi-Fi Docking 시스템을 구성하는 3가지 역할이 수행될 수 있는 실제 장치들의 예를 나타내었다.

이러한 구성으로 Wi-Fi Docking 시스템이 구축되면, 사용자는 Dockee를 마치 데스크탑 환경과 동일하게 사용할 수 있다. 하지만 만약 Peripheral 장치와 Docking Center의 인터페이스가 Bluetooth 기술을 사용할 경우, 대항 Peripheral 인터페이스를 Dockee로 Tunneling 하는 데에는 많은 기술적 제약이 따른다. 따라서 본 논문에서는 이러한 문제를 해결하기 위하여 Wi-Fi Docking 시스템의 새로운 기술들을 설계하였다.

3.2 Bluetooth HCI Tunneling 제안방식

일반적으로 Docking 시스템에서의 Dockee는 스마트폰 등의 소형 장치일 확률이 높으며, 대부분의 Docking Peripheral 장치는 Docking Center 장치에 USB를 통하여 연결되어 있는 경우가 많기 때문에, USB 인터페이스를 무선으로 전송해주는 WSB 기술을 사용하여 Dockee와 Peripheral 장치들을 간접적으로 연결할 수 있다. 다만, Docking Center와 Docking Peripheral 장치가 Bluetooth를 통하여 연결되어 있는 경우에는 그림 5와 같이 Tunneling을 지원한다. 또한 일반적으로 Bluetooth 장치는 S/W 영역이 탑재된 Host 부분과 MAC/PHY 등의 Chipset 영역인 Controller로 물리적인 구분이 가능하며, 이때 Host와 Controller의 물리적/논리적 통신 형태를 HCI(Host Controller Interface)로 구분한다. 본 고에서는 위와 같은 배경을 토대로, Dockee와 직접적으로 연결되지 않은 Bluetooth Peripheral 장치를 Dockee에 인식시키기 위하여 다음과 같이 제안한다.

Dockee가 Docking Center와 연결된 후 기존에 Docking Center와 연결된 Bluetooth 장치를 사용하고 자 할 경우에는, Dockee 및 Docking Center는 자신의 Bluetooth Host와 Controller의 HCI 연결을 각각 논리적으로 차단한다. 또한 Dockee와 Docking Center

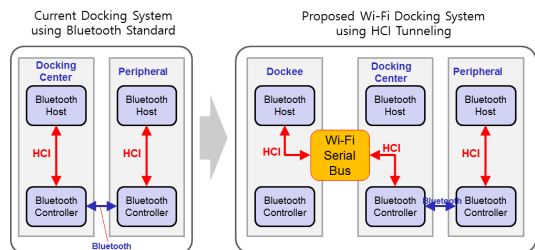


Fig. 5. Bluetooth HCI Tunneling via Wi-Fi Serial Bus.

는 Wi-Fi를 통하여 서로 연결한 뒤, WSB 연결을 완료한다. 그 후, Docking Center는 Bluetooth Controller의 HCI 정보를 WSB로 Encapsulation 한 뒤 WSB를 통하여 Dockee로 Forwarding 한다. 한편, Dockee 장치는 WSB로 Encapsulation 되어 있는 Docking Center의 Bluetooth Controller HCI 정보를 자신의 Bluetooth Host와 연결한다.

위와 같은 절차를 활용하면, Dockee의 Bluetooth Host는 Docking Center의 Bluetooth Controller와 논리적으로 연결되기 때문에, Dockee는 해당 Docking Peripheral 장치가 자신에게 직접 연결된 것처럼 인식하고 사용하게 된다.

기존의 Wi-Fi Docking 시스템 환경에서는 Dockee와 Docking Center의 연결과 Docking Center와 Peripheral 간의 연결에 대하여 각각의 Bluetooth LE가 필요하고 이에 대한 Controller 및 Profile이 3가지 장치에 모두 필요하다. 이 경우 Docking Center는 자신과 연결된 Peripheral 장치의 데이터를 Controller로부터 전달받아 자신의 Profile에 저장한 후, 이를 또 다시 Controller를 통하여 Dockee에게 포워딩해야 하는 단점이 있다. 또한 일반적인 Dockee 장치가 다수의 Bluetooth Peripheral 장치를 사용해야 한다는 환경적 시각에서 볼 때, 이러한 문제점은 많은 수의 Peripheral 장치 처리에 대한 복잡성이 매우 증가하기 되고, 이로 인하여 각각의 Peripheral 장치가 최종적으로 Dockee 장치에 접속되기까지 높은 접속 지연시간이 요구된다.

하지만 본 방식을 사용하면 Docking Center는 단지 WSB를 통하여 Controller 계층에서 그대로 Dockee에게 전달할 수 있기 때문에 응답속도가 높아질 수 있다. 또한 Dockee 측면에서는 자신과 Peripheral 사이에 존재하는 Docking Center의 존재를 Bluetooth 측면에서 알 필요가 없으며, Dockee는 해당 Peripheral 장치가 마치 자신과 직접 연결되어 있다고 판단하기 때문에 연결이 간소화되고 이로 인한 다수의 Peripheral 장치의 접속 시간이 단축될 수 있으며, 부가적으로 상호 운용성 및 호환성 측면에서 높은 효과를 얻을 수 있다.

IV. Tunneling Process

4.1 Host-Controller Synchronization

본 절에서는 Dockee와 Docking Center간 Docking 연결이 시도되기 전에, Docking Center와 Peripheral 장치가 Bluetooth를 통하여 이미 Pairing 되어 있다는

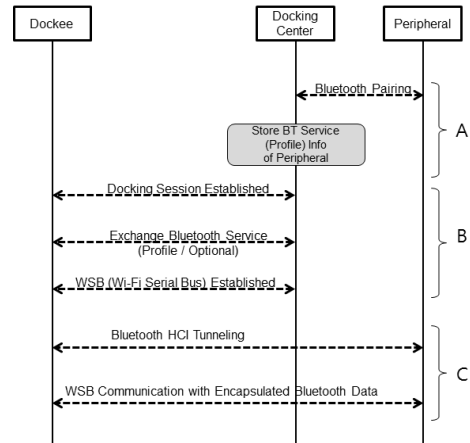


Fig. 6. Host-Controller Synchronization.

사실을 가정하고, 이를 통하여 Bluetooth Host 역할을 담당하는 Dockee 장치와 Bluetooth Controller 역할을 담당하는 Docking Peripheral 장치의 동기화 방법을 제안한다.

우선 A단계의 Docking Center 장치는 Peripheral 장치와의 Bluetooth Pairing 단계에서 획득한 Service 정보를 저장한다. 또한 B단계에서 Dockee 장치가 Wi-Fi를 통하여 Docking Center와 Docking Connection을 수행한 다음, Docking Center는 저장된 Bluetooth 장치의 Service 정보를 Dockee에 제공한다. 이때 Service 정보는 Peripheral 장치가 제공하는 Bluetooth Profile 정보를 포함하며, 물론 불필요할 경우 Service 정보 제공 절차를 생략할 수 있다.

C단계에서 Dockee 장치는 Docking Center의 Bluetooth Controller HCI 정보를 Wi-Fi Direct 기반의 WSB 기술을 이용하여 Docking Center와 Connection 절차를 완료한다. 이후 절차는 3.2절에서 나타낸 절차와 동일하게 Docking Center의 Bluetooth Controller HCI 정보를 WSB 통신을 활용하여 Dockee의 Bluetooth Host로 forwarding 하는데, 이때의 HCI 정보는 WSB를 통하여 Encapsulation 된 Bluetooth HCI 정보가 된다.

Bluetooth HCI Tunneling을 Encapsulation 하여 WSB로 Tunneling 하는 본 방식은, Dockee 장치의 Bluetooth Host와 Docking Center의 Bluetooth Controller를 논리적으로 연결하는 형태를 의미한다.

4.2 Service Discovery for Bluetooth

본 절에서는 Dockee 장치와 Docking Center 장치의 Service Discovery 절차를 통하여, 지원 가능한

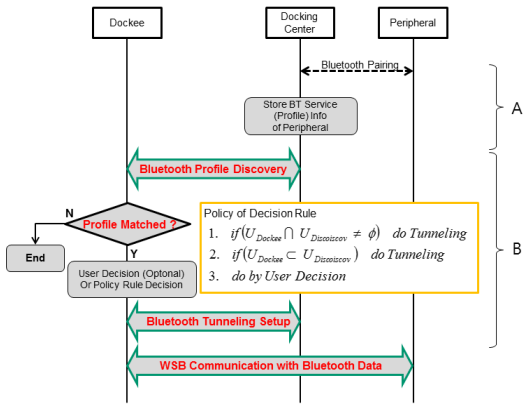


Fig. 7. Indirected Service Discovery Process.

Profile Matching을 기반으로 Bluetooth Tunneling 기능에 대한 사용 여부를 선택하는 방안을 설명한다. 먼저 그림 4와 같이, A단계에서는 Dockee 장치가 해당 Docking Center에 Docking Connection을 수행한 후, Docking Center에 종속적인 Bluetooth Peripheral 장치가 있을 경우 해당 Bluetooth 장치가 제공하는 Profile 등의 Service를 Discovery 한다.

B단계의 Profile Discovery 수행 과정에서는 Dockee의 Bluetooth Host의 Profile과 Discover된 Profile을 서로 Matching 시키는 작업을 수행하는데, 이때 각 Profile의 수가 복수일 경우가 존재한다. 예를 들어 Dockee의 Bluetooth Host가 Profile A, B, C를 제공하고, Discover 된 Peripheral 장치 (또는 Docking Center의 Bluetooth Controller)가 Profile B, C, D를 제공할 경우, Dockee가 사용할 수 있는 Docking Peripheral 장치의 기능은 상호 Matching 된 Profile의 공통 영역인 B와 C로 단축된다. 즉, Dockee는 Peripheral 장치의 Bluetooth Profile B와 C를 사용할 수 있다.

이 경우 Dockee는 특정 Policy에 의하여 해당 Bluetooth Peripheral 장치 사용 유무를 선택할 수 있는데, 이러한 Policy는 다음과 같은 3가지 형태로 구분된다.

- 두 장치 간 Matching된 Profile이 단 하나라도 존재할 경우 Bluetooth Tunneling을 시도
- Dockee의 모든 Profile이 Discover된 장치에 모두 포함된 경우에만 Tunneling 시도
- Matching 결과를 사용자에게 제공하고 사용자 결정에 의존하여 Tunneling 시도

이때 만약 Matching 된 Profile이 존재하지 않을 경우, Dockee는 해당 Bluetooth Peripheral 장치의 사용

을 포기한다.

이와 같은 절차는 모두 Docking Peripheral 장치들이 사전에 Docking Center 장치에 물리적으로 연결된 상태에서 Dockee 장치가 접속된다는 가정을 기반으로 설명되었다. 하지만, 사용자가 Docking 시스템을 사용하는 도중에 별도의 주변장치 연결을 시도하는 경우가 발생될 수 있다. 이 경우 Dockee 장치는 이미 Docking Center와 접속되어 사용하는 상태이며, 새로운 Bluetooth 장치의 Controller가 제공하는 새로운 Bluetooth Profile 연결 요구가 발생된다. 이러한 문제를 해결하기 위하여 본 고에서는 그림 8과 같은 절차를 제시한다.

그림 8은 Dockee와 Docking Center간 Docking Connection이 완료된 이후 새로운 Docking Peripheral 장치가 Docking Center에 Bluetooth Pairing을 수행할 경우에 대하여, Dockee가 해당 Peripheral 장치를 수용하는 절차를 나타낸 예시이다. 먼저 A단계에서 Dockee는 자신의 Bluetooth Host에서 제공하는 Profile 정보를 Wi-Fi ASP(Application Service Platform)를 통하여 Docking Center로 전달한다. 이때 Wi-Fi ASP 동작은 필요에 따라 UPnP Action 으로 대체될 수 있다.

이러한 절차를 통하여 Docking Center 장치는 Dockee 장치의 Bluetooth Profile 정보를 저장하며, 또한 Dockee와 Docking Center간 WSB Connection 과정을 수행한다. 이후, B단계에서 Docking Peripheral 장치가 Docking Center와 Pairing을 수행하기 위하여 Bluetooth Service Discovery 절차를 진행할 경우, Docking Center는 자신이 아닌 사전에 저

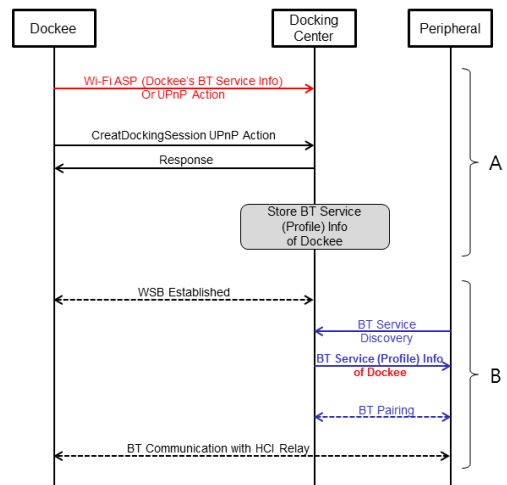


Fig. 8. New Bluetooth Tunneling after Dockee Connection.

장된 Dockee의 Bluetooth Profile 정보를 활용하여 Peripheral과의 Bluetooth Service Discovery 절차를 Dockee 장치를 대신하여 수행한다.

위 절차를 통하여 Docking Peripheral 장치는 최종적으로 Dockee 장치가 지원하는 Bluetooth Profile 정보를 활용하여 최종적으로 Pairing을 완료하게 되며, 또한 WSB Encapsulation을 통하여 Dockee는 Docking Peripheral 장치와의 Bluetooth Tunneling을 완료할 수 있다.

그림 9는 Docking Center와 Docking Peripheral 장치가 Bluetooth Pairing이 완료된 후에 Dockee 장치가 Docking Connection을 시도하는 경우에 대한 예시이다. 이 경우, A단계에서 Docking Center 장치는 이미 Peripheral 장치와의 Bluetooth Pairing 절차에서 습득한 Peripheral 장치의 Bluetooth Profile 정보를 저장한 상태이다. 따라서 B단계에서 Dockee 장치가 Docking Center에 Docking Connection을 시도할 경우, Dockee는 Docking Center에게 Wi-Fi ASP 또는 UPnP Action을 통하여 Docking Center에게 Peripheral 장치의 Bluetooth Profile 정보를 요청할 수 있다. 이때 Docking Center는 이미 저장된 Peripheral 장치의 Bluetooth Service 정보를 Wi-Fi ASP Response 형태나 GENA event notification, 또는 UPnP Response의 형태로 Dockee에게 제공한다. 이후 Dockee와 Docking Center 간에 Docking Session과 WSB 연결이 완료되면 기존과 동일한 형태로 Bluetooth HCI 정보를 Tunneling 하게 된다.

하지만, 이때 C단계에서의 Docking Center와 Peripheral 장치의 Bluetooth Pairing은 Peripheral 장치의 Bluetooth Host와 Docking Center의 Bluetooth

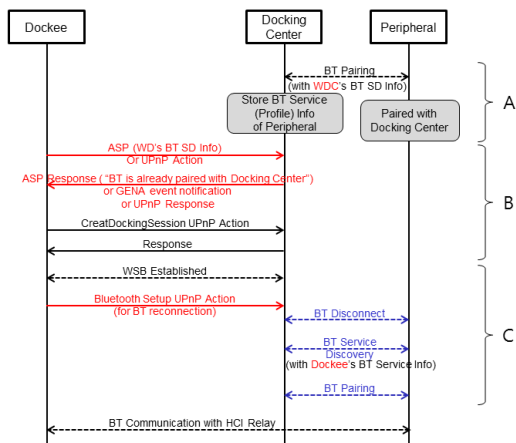


Fig. 9. Example of Bluetooth Pairing and Service Discovery before Dockee Connection.

Host 내에 존재하는 Bluetooth L2CAP(Logical Link Control and Adaptation Protocol)에 의하여 이미 전송계층 단계의 논리적 Session이 완료된 단계이기 때문에, 새로운 Dockee의 Bluetooth Host 내의 L2CAP과는 새로운 연결이 형성되어야 한다.

본 절차에서는 이러한 L2CAP 재접속 요구를 Bluetooth Pairing 자체의 재접속을 통하여 해결하는 형태를 그림 10에 나타내었으며, 이를 위하여 Dockee 장치는 Docking Center에게 Bluetooth Pairing 재접속을 요청하는 Bluetooth Setup Action 메시지를 송신한다. 또한 Dockee가 Undocking 할 경우, Docking Center와 Peripheral 역시 Bluetooth Pairing 절차를 다시 수행한다.

그림 10은 Docking Center와 Docking Peripheral 장치가 Bluetooth Pairing 절차를 완료된 후에 Dockee가 Docking Connection을 시도하는 경우에 대한 예시이다. A단계에서 Docking Center 장치는 이미 Docking Peripheral 장치와의 Bluetooth Pairing 절차에서 습득한 Peripheral 장치의 Bluetooth Profile 정보를 저장한다. 이후 B단계에서 Dockee 장치가 Docking Center에 Docking Connection을 시도할 경우, Dockee 장치는 Docking Center에게 Wi-Fi ASP 또는 UPnP Action을 통하여 Docking Center에게 Peripheral 장치의 Bluetooth Profile 정보를 요청한다. 이때 Docking Center는 이미 저장된 Peripheral 장치의 Bluetooth Service 정보를 Wi-Fi ASP Response, GENA event notification, 또는 UPnP Response의 형태로 Dockee에게 제공한다. 이후 Dockee와 Docking Center 간에 Docking Session과 WSB 절차가 완료되면 Bluetooth HCI 정보를 Tunneling 하게 된다.

하지만, B단계 이후 Docking Center와 Peripheral 장치의 Bluetooth Pairing은 Peripheral 장치의

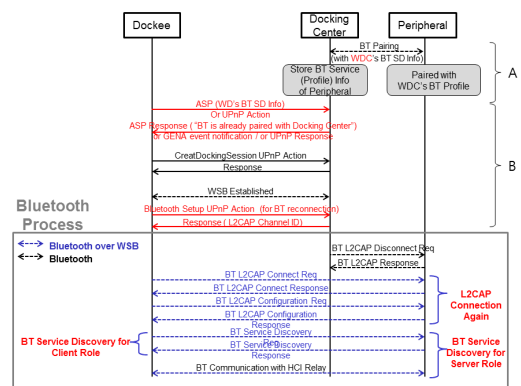


Fig. 10. Bluetooth Pairing Process after Dockee Connection.

Bluetooth Host와 Docking Center의 Bluetooth Host 내에 존재하는 Bluetooth L2CAP에 의하여 이미 전송 계층 단계의 논리적 Session이 완료된 단계이기 때문에, 그림 9에서 설명한 바와 동일하게 새로운 Dockee의 Bluetooth Host 내의 L2CAP과는 새로운 연결이 형성되어야 한다. 그러나, 본 절차에서는 앞서 기술된 바와 달리, Docking Center와 Peripheral 장치의 Bluetooth Pairing을 그대로 유지하면서, L2CAP Channel만을 재설정한다.

Dockee 장치로부터 Bluetooth Setup Action을 수신한 Docking Center 장치는 우선 기존의 Docking Peripheral 장치의 L2CAP Channel을 종료하기 위하여 Bluetooth L2CAP 종료절차를 수행한다. 또한 기존의 Docking Center와 Peripheral 장치의 Bluetooth Pairing을 그대로 유지하면서, Peripheral 장치와 Dockee 간의 Bluetooth L2CAP Channel을 새롭게 설정하기 위하여 Dockee는 WSB를 통하여 Peripheral 장치에게 Bluetooth L2CAP Connect를 요청하며, Docking Center는 이에 대한 응답으로 자신의 Bluetooth Host 내에 저장되어 있는 L2CAP Channel ID를 Dockee에게 전달한다.

이와 같은 방법으로 양 측의 Bluetooth Host 내의 새로운 L2CAP Connection을 설정하고, 그 후 Dockee가 필요한 Profile을 Peripheral 장치와 Matching시키기 위하여 WSB를 통한 Bluetooth Service Discovery 절차를 추가적으로 수행한다. 이 경우 Dockee의 Bluetooth 장치는 BT Service Discovery의 Client Role을 수행하고 Peripheral 장치는 BT Service Discovery의 Server Role을 수행한다. 또한 Dockee 장치가 Docking 세션을 종료할 경우, Peripheral 장치의 Bluetooth L2CAP Channel은 다시 Docking Center와 연결되어야 하므로, 본 절차에서 설명한 Bluetooth L2CAP 접속 절차를 역으로 수행한다.

V. 성능 평가

본 논문에서 제안한 Wi-Fi Docking 시스템의 Bluetooth Tunneling 기법은 Bluetooth 인터페이스를 요구하는 Docking Peripheral 장치를 현재 개발중인 Wi-Fi Docking 표준 기술에 그대로 접목시켜 적용할 수 있는 방안이다. Bluetooth 기술은 현재 무선 헤드셋과 무선 마우스 등에서 활발히 사용되고 있으며, 시장 영역이 점진적으로 확장되고 있기 때문에 Docking 시스템 환경에서의 고려가 반드시 필요하다. 또한 Bluetooth LE 기술 역시 IoT 테마와 함께 최근 수 많

은 소형 Accessory에서 활용되고 있으며, IoT 환경을 고려할 때 이 장치들은 스마트폰 또는 Laptop 등의 Dockee 장치들과의 Connectivity가 반드시 요구됨이 자명하다. 이에 본 고에서는 Bluetooth 및 Bluetooth LE를 활용한 Wi-Fi Docking 시스템의 접속 시간을 시뮬레이션 기반의 실험을 통하여 측정하고 그 결과를 기술하였다. 본 실험은 IEEE 802.11n을 대상으로 NS-2 (Network Simulator 2)를 통하여 진행되었으며, Dockee, Docking Center, Peripheral 모두 Bluetooth 4.0 표준을 반영하였다. 또한 802.11n 기능 중 Channel Bonding 기능을 배제하고 2.4GHz 대역만을 사용하여 진행하였다.

그림 11은 Bluetooth 인터페이스를 사용하는 Docking Peripheral 장치 수의 증가에 따른 Wi-Fi Docking 시스템의 Dockee 장치 접속 시간을 측정하여 나타낸 실험 결과이다. 이 결과에서의 Docking 접속 시간은 Dockee 장치가 Docking Center와의 접속 시간을 포함하여 전체 Docking Peripheral 장치 사용이 가능한 시점까지의 소요시간을 의미하며, 본 논문에서 제안한 Bluetooth Tunneling 기법을 활용하여 측정되었다.

또한 Bluetooth Tunneling Method 1은 Docking Peripheral 장치가 사전에 Docking Center 장치와 Bluetooth 연결이 되어있는 환경에서 Dockee 장치가 Docking Center와 연결되는 시나리오를 의미하며, 3.4절의 그림 9 및 그림 10에서 나타난 방식을 적용하여 동작된 실험 환경이다. 그리고 Bluetooth Tunneling Method 2는 3.4절의 그림 8에서 나타난 바와 같이 Dockee 장치가 Docking Center와 연결된 후에 새로운 Docking Peripheral 장치가 Bluetooth 인터

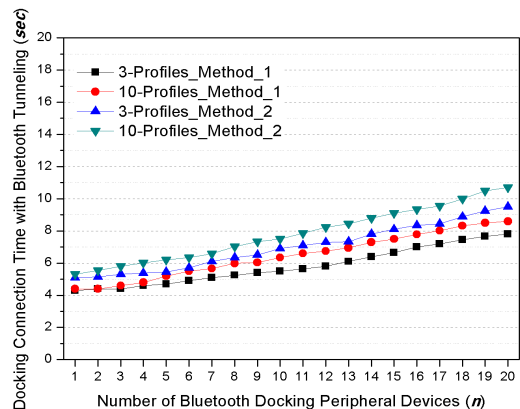


Fig. 11. Totally Consumed Wi-Fi Docking Connection Time including Bluetooth Tunneling Function.

페이스로 연결된 실험 환경을 의미한다.

본 실험 결과와 같이 Docking Peripheral 장치의 Bluetooth 연결은 Method 1 및 Method 2의 접속 유형에 따라 크게 달라지지 않으며, 각 Peripheral 장치가 지원하는 Profile 수 역시 크게 성능에 영향을 미치지 않는다는 결과를 나타내었다. 다만 연결을 시도하는 Bluetooth Docking Peripheral 장치의 수가 많을수록 전체 접속 시간에 영향을 미치는데, 이 결과 또한 Wi-Fi Docking Connection 완료 관점에서 크게 영향을 미치지 않는다. 즉, Wi-Fi Direct 연결 시간이 전체 성능의 가장 큰 요인이 되며, Bluetooth Tunneling 기능의 부가적인 영향은 사용자 환경 관점에서 무리가 되지는 않는 것으로 확인되었다.

그림 12는 Bluetooth LE 장치에 대한 실험을 그림 11과 동일한 환경으로 진행한 결과이다. 이 결과에서도 그림 11에서 나타낸 Bluetooth 연결 환경과 성능이 크게 다르지 않는 것으로 측정되었으며, 기존 실험 결과와 비교할 때 접속 시간 증가량의 기울기가 더욱 높은 점을 확인하였다. 이는 Bluetooth LE 기술이 기존 Bluetooth 기술 대비 짧은 접속 시간을 제공하기 때문에 나타난 결과이며, 기존 Bluetooth Pairing 단계에서 수행되는 전체 Channel 대상의 Discovery 절차가 Bluetooth LE 에서는 단지 3개의 Advertising Channel로 축소되었기에 가능한 것으로 판단되었다.

하지만 그림 12의 결과 중 Docking Peripheral 장치의 수가 많을 경우의 전체 Wi-Fi Docking Connection 시간은 그림 11에 비하여 증가된 결과가 측정되었다. 이러한 결과는 Bluetooth LE 기술이 기존 Bluetooth 기술 대비 낮은 Data Rate 및 매우 낮은

Throughput 성능에 따라 도출된 것으로 사료된다. 즉, Bluetooth LE 기술은 기존 Bluetooth 대비 매우 높은 Sleep State 시간 점유율을 토대로 Duty Cycle이 운용되고 있으며, 이는 저전력 프로토콜 특성 상 나타날 수 있는 결과로 판단된다.

VI. 결 론

본 고에서 제안한 기술은 향후 발표될 Wi-Fi Docking 표준 기술에서 기존 Bluetooth 및 Bluetooth LE 인터페이스를 탑재한 주변 장치들을 그대로 Docking Peripheral 장치로 활용할 수 있기에 향후 적용성이 매우 높게 예측되며, 또한 Bluetooth 표준 및 Wi-Fi Direct를 포함한 향후 발표될 Wi-Fi Docking 표준 기술의 변경 없이 그대로 적용이 가능하기 때문에 확장성 또한 높게 판단된다. 본 기술을 시작으로, 상호 기술적 협력이 어려운 이기종 표준 기술들의 interoperability 측면을 강화한 현실적인 방안들을 추가적으로 연구할 계획이며, 향후 IoT 환경에서 동일 주파수 대역을 사용하는 수 많은 이기종 통신 기술들의 Coexistence 방안 또한 본 논문의 제안 기술과 관련하여 더욱 깊이 연구할 계획이다.

References

- [1] *Wi-Fi Docking Technical Specification Draft (V0.15)*, Wi-Fi Alliance, 2015
- [2] *Wi-Fi Peer-to-Peer (P2P) Technical Specification v1.2*, Wi-Fi Alliance, 2010.
- [3] IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band, IEEE Std 802.11ad, 2012.
- [4] *WiGig Specification v1.0*, Wi-Fi Alliance, 2009.
- [5] *Wi-Fi Display Technical Specification v1.1*, Wi-Fi Alliance, 2012.
- [6] *Wi-Fi Serial Bus Technical Specification*, Wi-Fi Alliance, 2015.
- [7] *Bluetooth Core Specification 2.1*, Bluetooth

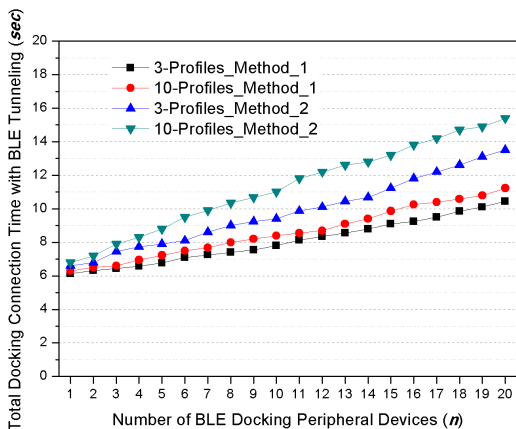


Fig. 12. Totally Consumed Wi-Fi Docking Connection Time including Bluetooth Tunneling Function.

- SIG, Jun. 2007.
- [8] *Bluetooth Core Specification 4.2*, Bluetooth SIG, Dec. 2014.
- [9] *Marketing Requirements Document for Interoperability Testing of Wi-Fi Docking*, Wi-Fi Alliance, 2013.
- [10] J. Lee, "A new routing scheme to reduce traffic in large scale mobile ad-hoc networks through selective on-demand method," *Wirel. Netw.*, vol. 20, no. 5, pp. 1067-1083, 2014.
- [11] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with wi-fi direct: Overview and experimentation," *IEEE Wireless Commun.*, pp. 96-104, 2013.
- [12] H. Joh and I. Ryoo, "A hybrid Wi-Fi P2P with bluetooth low energy for optimizing smart device's communication property," *Peer-to-Peer Networking and Appl.* 8.4, pp. 567-577, 2015.

이 재 호 (Jaeho Lee)



2005년 : 고려대학교 전자컴퓨터공학과 석사
2008년~2013년 : 고려대학교 전기전자전파공학과 박사
2011년~2013년 : 서원대학교 겸임교수
2013년~2015년 : LG전자 차세대표준연구소 선임연구원
2015년~현재 : 서원대학교 정보통신공학과 조교수
<관심분야> WPAN, 센서네트워크, MANET, MAC, WBAN, Bluetooth, Wi-Fi, ITS, Localization