논문 21-46-10-18 The Journal of Korean Institute of Communications and Information Sciences '21-10 Vol.46 No.10
https://doi.org/10.7840/kics.2021.46.10.1698

5G 단말 소프트웨어 모뎀의 셀 탐색 구현을 위한 Finite State Machine 기반 프로토콜 소프트웨어 구조 설계

김 혜 원*, 김 주 엽°

A Design of Protocol Software for Implementing Cell Search of 5G User Equipment Software Modem Based on a Finite State Machine

Hyewon Kim*, Juyeop Kim°

요 약

소프트웨어를 중심으로 무선통신 시스템을 구현하는 소프트웨어 모뎀이 현재 주목을 받고 있다. 이 소프트웨어 모뎀을 실현하기 위해서는 소프트웨어로 구현된 알고리즘의 계산 효율성 개선과 함께 이런 알고리즘을 유연하게 구동할 수 있는 개선된 프로토콜 구조가 필요하다. 특히 프로토콜의 경우 다양한 무선통신 시나리오에 대응해서 구현하기 쉬운 형태의 소프트웨어 구조가 있어야 향후 프로토콜에 대한 유지보수가 용이하다. 이에 본 연구에서는 5G cell search에 대한 프로토콜을 구현할 수 있는 소프트웨어 구조를 제안한다. Cell search는 5G 단말의 성능에 중요한 요소에 해당하며, 단말의 상황에 따라 다양한 시나리오가 발생하므로 유연한 프로토콜 소프트웨어 구조를 필요로 한다, 본 연구에서는 Finite State Machine(FSM) 개념을 사용하여 cell search 에 대한 프로토콜을 설계하였으며, 제안된 프로토콜 설계의 성능 검증을 위해 USRP 기반의 실제 시스템 환경에서 구현에 대한 평가를 관련 선행연구인 OpenAirInterface 코드와 비교 분석을 하였다.

키워드 : 소프트웨어 모뎀, 프로토콜 설계, 5G, 이동통신, Finite State Machine
Key Words : communication, signal processing, Neutral systems, Communication Sciences, Network

ABSTRACT

Software modem, which implements whole wireless communicaiton system by software, is currently focused on. To realize this software modem, it is needed to improve efficiency of algorithm computation by software, as well as to design the software architecture of the corresponding protocol which can flexibly operate the algorithms. Especially the protocol needs to be flexible in order to deal various scenarios and to make itself to be maintained easily. In this circumstance, this research focuses on designing a flexible software architecture for 5G cell search. Cell search is an important factor in view of the performance of 5G mobile terminals and requires to cover various scenarios with a flexible architecture. In this research, we utilize the concept of Finite State Machine (FSM) to design the protocol, and perform performance evaluation by implementing it in a real-time operating system and comparing with OpenAirInterface.

※ 본 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2018-0-00726, Beyond 5G 개념의 Software-Defined Cell Beam Search 기술 개발)
* First Author : Contela, hwkim@contela.co.kr, 정회원
° Corresponding Author : Sookmyung Women's University, Department of Electronics Engineering, jykim@sookmyung.ac.kr, 정회원
논문번호 : KICS202106-136-D-RN, Received June 15, 2021; Revised July 24, 2021; Accepted July 27, 2021

Ⅰ. 서 론

이동통신의 급격한 발전에 유연하게 대처하기 위해 Radio Frequency(RF)를 비롯하여 모뎀 기능의 전 과정을 소프트웨어로 재정의 할 수 있는 소프트웨어 모뎀에 대한 연구가 활발하게 이루어지고 있다[1]. 기존 하드웨어 모뎀의 경우 계산량이 많은 알고리즘에 대해 전용으로 처리하는 하드웨어가 별도로 존재하는 구조를 가지며, 이로 인해 프로토콜은 알고리즘의 계산량에 대한 고려 없이 별도 개념으로 설계 및 구현이 가능하다. 하지만 소프트웨어 모뎀은 알고리즘 및 프로토콜이 하나의 하드웨어에서 모두 처리되는 구조를 가지므로, 알고리즘과 프로토콜 모두 제한된 계산 환경에서 성능을 극대화시킬 수 있는 구현이 필요하다.

이 소프트웨어 모뎀 기술이 상용화의 결실을 이루기 위해서는 현재 하드웨어 기반 모뎀에 비해 처리속도 면에서의 열화된 성능 특성을 보완하는 것이 중요하다. 현재의 소프트웨어 모뎀 관련 플랫폼이나 오픈소스 프로젝트가 대부분 주로 성능 테스트 용도로 국한되는 것은 기술적 관점에서는 이런 처리속도 성능으로부터 기인한 것이라 볼 수 있다[2-4]. 따라서 소프트웨어 모뎀이 기존 하드웨어 기반 모뎀의 역할을 대체하기 위해서는 하드웨어 기술의 발전도 필요하지만 관련 알고리즘의 계산 성능 향상과 더불어 프로토콜 수준에서도 알고리즘을 효율적으로 구동할 수 있도록 개선하는 것이 중요하다.

소프트웨어 모뎀의 프로토콜에 대한 기존 연구로 소프트웨어 모뎀으로 가장 대표적인 오픈소스 프로젝트인 OpenAirInteface(OAI)를 생각해볼 수 있다. 이 OAI의 소프트웨어 모뎀은 각종 송수신 신호 처리에 필요한 알고리즘이 모두 소프트웨어로 구현되어 있어 범용 하드웨어 자원만으로 5G 기지국 혹은 단말 역할을 수행할 수 있게 한다.[5] OAI의 오픈소스를 통해 실제 시스템 상에서의 실시간 처리 개념으로 cell search를 포함한 각종 알고리즘을 처리할 수 있다. 그러나 세부적으로 구현을 보면 각 프로세스와 프로토콜 계층 간의 역할 구분이 모호하게 구성되어 있어서, 특정 부분의 코드 영역이 지나치게 길어 가독성이 저하되거나 특정 계층의 역할 비중이 치우치는 등의 문제가 존재한다. 따라서 OAI 소스코드를 분석하거나 이를 기반으로 새로운 기능 추가 및 수정을 하는데 있어서 많은 어려움이 따른다. 이런 OAI 프로토콜의 특징은 소프트웨어 모뎀이 기본적으로 하드웨어 기반 모뎀에 비해 장점으로 가져야 하는 유연성을 저해하는 요소라고 볼 수 있다.

이를 개선하기 위해서는 소프트웨어 상에서 상위 및 하위 계층 및 한 계층 내 세부 개체에 대한 역할 구분을 명확하게 하는 구현이 필요하다. 또한 각 계층 및 세부 개체의 구동 상황과 이벤트를 고려하여 일어날 수 있는 모든 상황에 대해 유연하게 대처할 수 있는 코드 설계가 필요하다. 그런 취지에서 본 연구에서는 5G 시스템에서 cell search의 전체 과정을 소프트웨어 모뎀 개념으로 효율적으로 구현하기 위한 유연한 프로토콜 소프트웨어 설계를 제안하고자 한다. Cell search는 단말이 이동통신 망에 접속하기 위해 인접 기지국을 찾는 과정이다. 기본적으로 데이터 연결이 없는 초기 접속 과정에서 주변의 기지국에 대해 cell search를 수행한다. 하지만 이와 더불어 이동 환경에서의 데이터 연결 유지를 위해 현 기지국과 인접 기지국을 찾을 때도 cell search가 필요하다. 따라서 이동 환경에 있는 단말은 기본적으로 cell search 절차를 짧은 주기로 반복을 해야 한다. 소프트웨어 모뎀 개념에서는 이 cell search는 다른 송수신 처리와 병렬로 처리되어야 하므로, cell search에 대한 알고리즘을 효율적이면서 적시에 구동할 수 있는 유연한 프로토콜 설계가 필요하다.

본 논문의 2장에서는 3GPP의 5G 표준을 분석하고 cell search와 관련된 프로토콜 계층의 역할을 고려하여 시스템 모델을 제시한다. 이를 바탕으로 3장에서 cell search에 대한 프로토콜 설계 과정을 설명한다. Cell search의 service primitive를 주고 받는 상위 계층과 해당 cell search service primitive를 상황에 맞추어 처리할 수 있도록 cell search 신호처리를 Finite state machine(FSM) 구조로 설계한다. 4장에서는 제안된 프로토콜 구조를 실제 코드로 구현한 구조를 설명하면서 OAI와의 비교 분석을 하면서 구현된 코드 구조의 특징을 설명한다. 5장에서는 제안된 프로토콜 구조를 소스코드로 구현하고 실시간 시스템 환경에서의 동작 성능을 실험을 통해 검증하면서 제안된 프로토콜 설계에 대해 정량적 관점에서 분석하고 6장에서 결론을 내린다.

Ⅱ. 시스템 모델

2.1 Cell Search 용 소프트웨어 모뎀 구조

그림 1은 cell search 구현을 위한 전체 소프트웨어 모뎀 구조를 나타내고 있다. 기존 하드웨어 기반 모뎀에서의 신호처리 알고리즘은 독립적인 하드웨어에서 처리되기 때문에 프로토콜의 동작과 명확히 분리되는 반면, 소프트웨어 모뎀에서는 그림 1의 구조와 같이

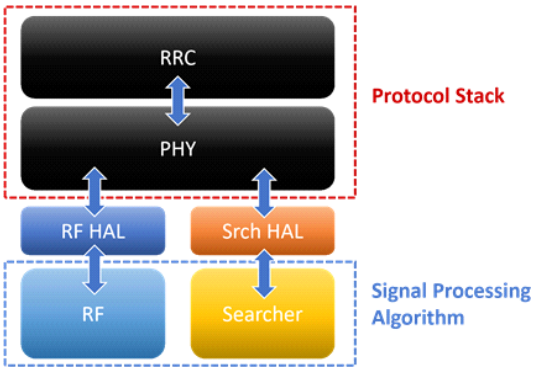

그림 1. 전체 Cell search 시스템 모델
Fig. 1. Cell Search System Model

알고리즘과 프로토콜 전체가 한 소프트웨어 바이너리 내에 존재한다. 따라서 알고리즘과 프로토콜 관련 소프트웨어 부분의 코드 영역의 구분을 위해 구조적으로 Hardware Abstraction Layer(HAL)가 그 사이에 존재하는 모델을 가정하였다. 소프트웨어 모델은 알고리즘과 프로토콜의 경계가 모호하게 될 수 있으며, 이 경우 서로 영역을 침범했을 때 심각한 문제가 발생할 수 있다. 따라서 서로 직접적인 영향을 주지 않도록 HAL이 중재하는 구조를 가정하였다.

그림 1 하단에 존재하는 RF와 searcher는 각각 RF와 cell search의 신호처리에 필요한 알고리즘을 처리하는 부분이며, 소프트웨어 모델을 가정하는 본 논문에서는 이 부분들은 하드웨어가 아닌 실시간 소프트웨어로 구현되어 동작하는 것을 가정하였다. HAL은 각 알고리즘과 별도로 존재하며 프로토콜은 이 HAL을 통해서만 알고리즘 동작을 제어하거나 데이터에 접근할 수 있다고 가정하였다. RF와 searcher 알고리즘은 각각 독립적인 thread에서 동작하며, searcher는 동시에 여러 cell에 대한 탐색을 수행하기 위해서 여러 개의 thread로 병렬 구동될 수 있도록 이루어져있다고 가정하였다. 프로토콜은 상황에 맞게 알고리즘의 thread를 적절히 제어하여 발생한 정보들을 이용하여 cell search 신호처리를 진행한다.

Cell search역할을 하는 프로토콜은 Layer 3 개념에 해당하는 Radio Resource Control(RRC)와 Layer 1에 해당하는 PHY로 나뉜다. Cell search 과정에서 필요한 역할들이 이 RRC 및 PHY 부분에 의해 구분되어 수행되며, 다양한 시나리오를 유연하게 구현하기 위해서는 RRC와 PHY 간 직접적인 접근 및 제어가 이루어지지 않도록 코드 관점에서도 분리되어 구현된다고 가정하였다. 즉, 프로토콜의 최하위 계층인 PHY에서는 RRC에서 전달된 service primitive에 따라 적절한 HAL에서 제공하는 함수를 호출하여 알고리즘을 제어할 수 있다.

2.2 Cell Search 관련 알고리즘

그림 2에서는 5G에서 cell search를 한번 수행하는 알고리즘의 순서를 보여주고 있다. 첫 번째로 cell과의 시간 동기를 맞추기 위해 Primary Synchronization Signal(PSS) 탐지를 수행한다. 5G에서는 기본적으로 동기 신호를 20ms 마다 적어도 한 번씩 보내기 때문에 20ms 구간의 신호를 샘플링하고 PSS sequence에 대해 시간 축 상에서의 cross-correlation 결과를 비교한다. 샘플링 데이터 내에서 PSS 탐지에 성공하면 cell ID의 일부 정보에 해당하는 Nid2 값을 얻을 수 있다. 이후에 PSS에서 2 ODFM symbol만큼 떨어져 있는 Secondary Synchronization Signal(SSS)를 탐지한다. 우선 PSS가 포함된 수신 OFDM symbol을 기반으로 coarse 개념의 frequency offset 추정 및 보정을 수행한 뒤[6], PSS 탐지를 통해 파악된 시간 동기를 기반으로 Fast Fourier Transform(FFT) 과정을 수행하여 frequency-domain symbol 값들을 획득한다. 이후 과정인 SSS 탐지에서는 후보 SSS sequence들과의 correlation 결과 대조를 통해 가장 적절한 송신 SSS sequence를 찾아낸다. SSS detection까지 성공하면 PSS 및 SSS 탐지에서 도출된 Nid1,2 정보를 조합해서 cell ID를 알 수 있게 된다. 마지막으로 수신 신호가 cell의 동기 신호임을 확신하기 위해 Physical Broadcast CHannel(PBCH)에 대한 복호 과정을 수행한다. PBCH 복호에 성공하면 신호의 빔 번호에 해당하는 Synchronization Signal Burst(SSB) index 값을 얻을 수 있으며 이를 기반으로 frame의 시작 지점을 파악하면서 slot 및 frame 동기화를 마칠 수 있다.

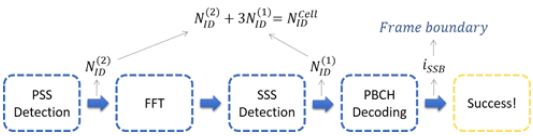

그림 2. Cell search 알고리즘
Fig. 2. Cell Search Algorithm

2.3 Cell Search 시나리오 요구사항

3GPP의 5G 표준 TS 38.304[7]에 따르면, 이전 cell 정보가 없는 단말은 최초로 cell을 선택하기 위해서는 모든 RF channel을 스캔해서 suitable cell을 찾아야 한다. Suitable cell이란 cell Selection criterion (S-criterion)이라는 특정 조건을 만족하여 단말이 머

무르기에 적합한 cell을 뜻한다. S-criterion을 만족하기 위해서는 SSB를 기반으로 cell의 신호 세기가 기지국에서 제어정보로 제공하는 특정 기준 값 이상이어야 한다. 따라서 단말은 suitable cell을 찾기 위해 우선 RF channel을 하나씩 스캔하여 해당 주파수 내의 가장 수신신호 세기가 강한 cell을 찾는다.

PSS detection 과정에서 동기 신호의 기본 주기인 20ms 내에서 PSS sequence와의 correlation값이 가장 높은 cell 후보를 택하며, 이를 가장 수신신호 세기가 강한 cell로 간주한다. 이 때 선택된 cell이 S-criterion을 만족하는지 확인하기 위해서 해당 cell search에서 측정된 Reference Signal Received Power(RSRP)와 함께 해당 cell에서의 S-criterion 기준 값을 수집하며, 이 정보들은 해당 cell이 보내는 System Information Block(SIB)로부터 얻는다. 이 SIB는 shared channel을 통해 수신이 되므로, 단말은 탐색된 cell과 지속적인 동기를 유지하면서 SIB를 수신해야 S-criterion 만족 여부를 파악하고 cell search 과정을 마칠 수 있다. 여기서 특정 cell에 대해 지속적으로 동기화를 수행하여 동기를 유지하는 상태를 cell tracking이라고 부르도록 한다. Cell tracking 상태에서 단말은 SIB정보를 수신하여 S-criterion 만족 여부를 확인하고 만족을 하는 경우 해당 cell을 최종적으로 탐색을 성공한 것으로 간주한다.

2.4 Cell Search 시나리오 도출

위의 요구사항을 만족하도록 cell search의 여러 가지 시나리오를 정의하면 다음과 같다. 우선 cell search의 모든 세부 과정이 정상적으로 마쳐지는 시나리오 흐름은 그림 3과 같다. PHY는 RRC가 내리는 service primitive에 따라 cell search 알고리즘을 구동하여 cell search를 실질적으로 수행하도록 한다. RRC가 PHY에 RF channel 정보를 포함하는 cell search command 을 내리면 PHY는 HAL 함수를 호출하여 cell search 알고리즘을 수행하고 찾은 cell 정보를 담은 report를 올린다. RRC는 PHY로부터 받은 report로부터 cell 정보 내용을 참고하여 이 cell에 머물러도 해도 되는지 판단한다.

이후 RRC는 해당 cell을 대상으로 cell tracking을 하도록 cell tracking command을 내린다. Cell tracking command를 받은 PHY는 RRC로부터 받은 cell 정보에 대해 cell search 알고리즘을 다시 수행하여 동기 정보를 다시 획득하고 이를 기반으로 해당 cell의 기지국 신호를 tracking하기 시작하면서 shared channel에 대한 수신 준비를 한다. 그리고 shared

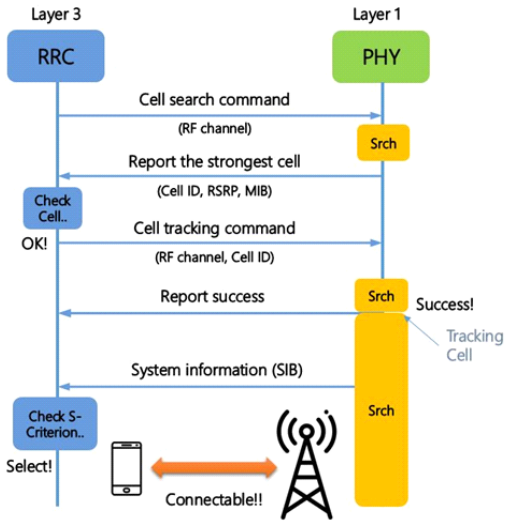

그림 3. RRC와 PHY간 Cell Search 성공 시나리오
Fig. 3. RRC-PHY Interoperations in Cell Search Normal Scenario

channel로부터 전달되는 SIB를 획득한다. RRC에서는 PHY에서 전달받은 SIB 내 정보를 이용하여 S-criterion 만족 여부를 판단하고, 그 기준을 만족하면 해당 cell을 selection하게 된다.

그림 4-6은 cell search에 실패하는 여러가지 시나리오를 나타내고 있다. 그림 4의 시나리오와 같이 cell search에 실패하면 RRC는 현재 대상으로 하는 RF

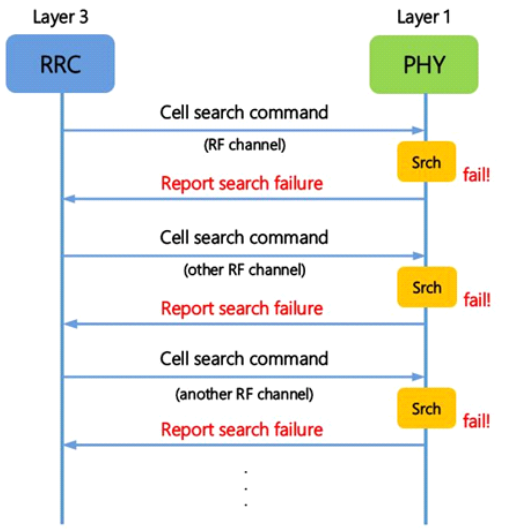

그림 4. RRC와 PHY간 Cell Search 실패 시나리오
Fig. 4. RRC-PHY Interoperations in Cell Search Failure Scenario

channel에 cell이 없다고 판단하고 다른 RF channel를 대상으로 PHY에 cell search command를 다시 내린다. RRC는 이와 같은 절차를 cell search 성공에 대한 report를 받을 때까지 반복한다.

그림 5의 시나리오는 주어진 RF channel에서 cell을 물리적으로 찾았으나 관련 정보를 확인 결과 접근이 금지(Barred)된 경우에 해당한다. Cell이 접근 금지된 경우에는 Master Information Block(MIB)내

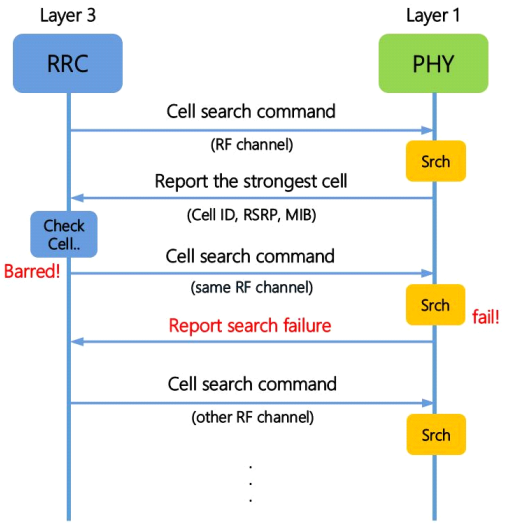

그림 5. RRC와 PHY간 Cell Barred 시나리오
Fig. 5. RRC-PHY Interoperations in Cell Barred Scenario

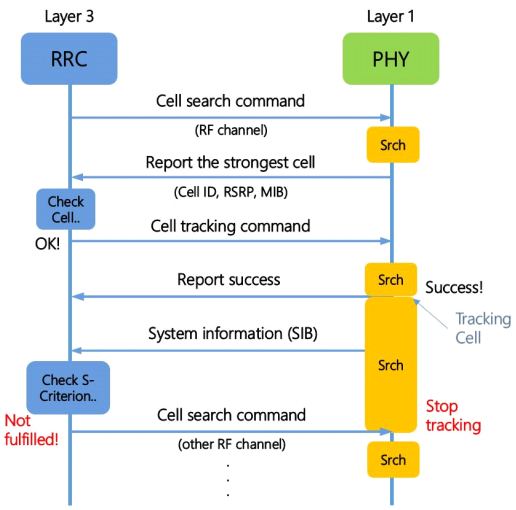

그림 6. S-criterion이 만족되지 않는 시나리오
Fig. 6. RRC-PHY Interoperations in Not Satisfying the S-criterion

CellBarred field가 'barred'인 경우와 MIB를 수신하지 못한 경우가 존재한다. 이렇게 cell search 결과로 cell이 barred 처리되는 경우, 해당 cell은 300초 동안 cell search 후보에서 제외되어야 하므로, RRC는 동일 RF channel에서 해당 barred cell을 제외한 다른 cell을 찾게 된다.

그림 6은 cell search에 성공하고 cell tracking 과정에서 S-criterion을 만족하지 못하는 경우이다. Cell tracking 과정에서 SIB 메시지를 획득하게 되며, SIB 내의 S-criterion 관련 파라미터들을 파악하게 된다. Cell search 알고리즘에서 측정된 RSRP 값과 SIB 내 파라미터 값들을 기반으로 S-criterion을 고려하였을 때 그 기준을 만족하지 못하면 해당 cell을 선택하지 못한다. 이 경우 단말은 해당 cell에 대한 tracking을 중지하고 다른 RF channel을 대상으로 cell search를 수행해야 한다.

Ⅲ. Cell Search 과정을 위한 PHY 프로토콜 설계

위 시나리오들을 대상으로 cell search 과정을 수행하는 RRC 및 PHY의 프로토콜을 설계하고자 한다. 기존 관련 연구에 해당하는 OAI에서는 cell search 기능이 PHY 계층 코드에서 cell search 전 과정이 한번의 직렬 흐름으로 처리되고 있다. 따라서 OAI는 계층 내 entity간 역할 분리 구조가 없으며 OAI 코드를 기반으로 새로운 기능을 추가하거나 기존 기능을 수정하기 까다롭다. 따라서 프로토콜 구현의 유연성을 확보하기 위해서는 OAI 코드와는 다르게 모듈화 구조의 프로토콜 설계가 필요하다.

또한 본 설계에서 대상으로 하는 cell search 시나리오는 초기 cell을 찾는 경우 뿐 아니라 cell tracking 도중에 인접 cell에 대한 measurement를 병행해서 수행되는 것도 포함할 수 있어야 한다. 따라서 어느 시점에서도 병렬적으로 cell search 알고리즘 처리가 될 수 있는 유연한 프로토콜 동작이 필요하다. 이를 위해서는 cell search 관련 신호처리를 세부적으로 나누어 진행되도록 FSM 구조로 설계될 필요가 있다. 또한 scheduler 개념을 도입하여 cell search의 병렬 구조 수행을 관리할 수 있도록 하고 PHY에서 다양한 시나리오에 대해 효율적으로 cell search 처리를 시작할 수 있도록 한다.

3.1 Cell Search Scheduler 설계

그림 7은 본 연구를 통해 설계된 scheduler 중심의

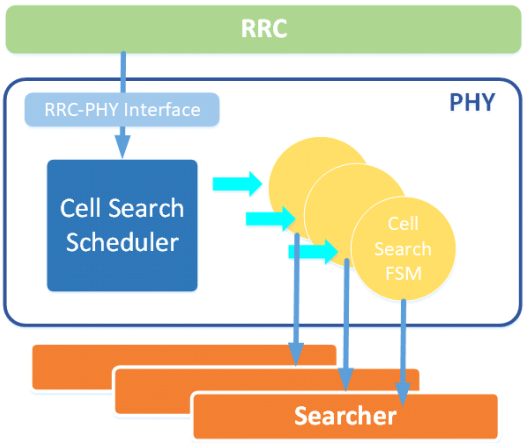

그림 7. Cell search scheduler 중심의 PHY 프로토콜 전체 구조
Fig. 7. Scheduler-Centric Structure of PHY Cell Search Protocol

PHY 프로토콜 구조를 보여준다. 여기서 cell search scheduler는 상위 계층에서 내려온 각종 cell search 관련 service primitive에 대응하여 cell search 신호처리를 scheduling하고 관리한다. Cell search scheduler는 프로토콜 관점에서 cell search 알고리즘을 유연하게 시작하게 하는 측면에서 cell search 프로토콜에 있어서 반드시 필요하다. Cell search scheduler는 다양한 cell search 시나리오에 의해 다발적으로 발생할 수 있는 cell search 요청을 효율적으로 처리하는 역할을 한다. 또한 기존에 고려되지 않은 새로운 cell search 시나리오를 유연하게 구현하기 쉽도록 cell search 동작을 구조적으로 수행하게 하는 역할도 한다.

Cell search scheduler에 의해 한번 수행하는 cell search 신호처리는 그림 8의 FSM과 같이 나타낼 수 있다. 제안된 FSM 구조는 한번의 cell search 신호처리가 특정 시간에 머무르는 상태를 state로 정의하고, 현재 state에서 발생하는 각 event에 따라 적절히 취할 action과 다른 state로의 천이를 기술하고 있다. Cell search scheduler는 cell search 건 별로 그림 8의 FSM을 구동하기 위한 thread를 생성하여 병렬 구조로 처리한다.

2장에서 도출된 시나리오 상에서 상위 계층이 service primitive를 통해 cell search를 시작하며, PHY는 전달된 service primitive의 유형에 따라 cell search가 필요한 경우 scheduler queue에 service primitive을 저장하고 scheduler를 통해 cell search가 적시에 수행되도록 한다. Cell search가 필요한 시점에서 scheduler는 cell search FSM을 시작하여 cell

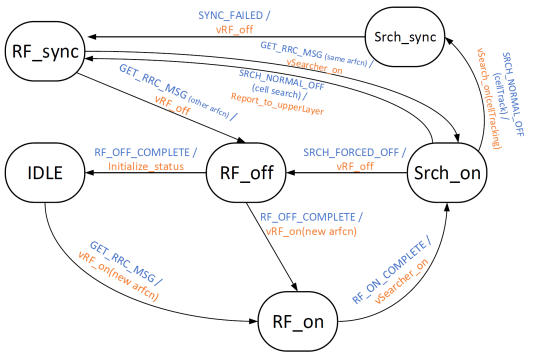

그림 8. Cell Search 에 대한 FSM 세부 구조
Fig. 8. Detailed FSM of the Cell Search Protocol

search 알고리즘을 1회 실행한다. 또한 service primitive가 cell tracking인 경우, cell search 알고리즘을 동일하게 수행한 뒤 지속적인 동기화를 위해 관련 cell tracking 알고리즘을 반복적으로 구동한다.

한 번의 cell search 신호처리 과정은 그림 8에 표현된 FSM 구조를 기반으로 동작한다. 초기에는 IDLE 상태에 머물러 있으며, 이 상태에서 FSM은 새로운 service primitive 등에 의해 cell search가 필요해지는 시기를 기다리고 있다. 그리고 scheduler에 의해 cell search가 시작되면 이 상태를 나오면서 RF 장치를 제어하는 RF 알고리즘을 구동하기 시작하고 RF_on 상태로 천이된다. RF 알고리즘이 정상적으로 구동되고 나서 FSM은 cell search 알고리즘 부분에 해당하는 searcher를 구동하기 시작하면서 Srch_on 상태로 천이된다.

Srch_on 상태에서는 searcher의 구동 결과에 따라 두 가지 동작 및 상태 천이가 발생할 수 있다. 우선 searcher가 cell을 발견할 경우 해당 RF channel에 cell이 존재하는 것을 의미한다. 이때 상위 계층이 곧바로 cell tracking에 대한 요청을 할 것이 통상적으로 예측되므로, FSM은 RF 알고리즘을 멈추지 않고 지속적으로 구동 상태에 두는 RF_sync 상태로 천이된다. Searcher 가 cell 찾기에 실패하는 경우에는 해당 RF channel에 cell이 없는 것을 의미하므로, RF 알고리즘을 종료하기 시작하면서 FSM은 RF_off 상태로 천이되고, RF 알고리즘이 완전히 종료되면 IDLE 상태로 회귀한다.

FSM이 RF_sync 상태에 있을 때는 상위 계층에서 내려오는 service primitive의 cell search 주파수 정보에 따라 두 가지 동작이 발생할 수 있다. 현재 RF 알고리즘이 구동하는 주파수와 service primitive의 cell search 주파수가 동일한 경우 RF 알고리즘 동작을 그

대로 유지하면서 searcher를 구동하는 형태로 동작한다(그리고 FSM은 Srch_on상태가 된다). 한편 RF 알고리즘이 구동하는 주파수와 cell search 주파수가 다른 경우 RF를 재설정해야 할 필요가 있으므로 FSM은 RF_off로 천이되면서 RF 알고리즘을 한번 종료한다. 그리고 새로운 cell search 주파수로 RF를 다시 구동시키면서 FSM은 RF_on으로 천이된다. 이후 원래 cell search 동작 과정과 동일하게 FSM은 searcher 구동과 함께 Srch_on으로 천이된다.

반면 RF_sync 상태에서 cell tracking에 대한 service primitive를 수신하게 되면 FSM은 우선 Srch_on 상태로 천이하면서 searcher가 동기정보를 재획득하기 위한 cell search를 수행하고, cell search가 성공적으로 끝나면 동기정보를 기반으로 searcher에게 cell tracking를 하도록 구동하면서 Srch_sync으로 천이된다. 그 이후에는 searcher가 cell tracking을 실패하기 전까지 그 상태를 유지한다.

3.2 RRC Layer 설계

2장에서 도출된 cell search 시나리오를 고려할 때, cell search 과정은 RRC와 PHY의 상호작용에 의해 진행되며, RRC로부터 전달되는 service primitive를 시작으로 모든 cell search 과정이 시작되어야 한다. OAI 코드에서는 이와 같은 계층의 역할분담과는 다르게, 실행 시점에서 search 주파수 값이 고정되고, 그 고정값이 PHY 관련 소스코드의 변수에 기입되는 형태로 전달된다. 이와 같은 코드 동작 구조는 cell search를 할 여러 주파수에 대해 고려하는 순서대로 순차적으로 수행하는 등의 구현이 어렵다. 따라서 유연한 프로토콜 구현을 위해서는 OAI 코드 구현과는 다르게, RRC에서 cell search를 할 주파수를 지정하고 이를 service primitive로 전달함으로써, PHY는 RRC에서 지정한 주파수에 대해 수동적으로 cell search 신호처리를 수행하는 형태로 구현 설계가 이루어져야 한다.

위와 같은 RRC의 역할을 고려해서 그림 9와 같은 RRC의 FSM을 설계하였다. RRC 또한 PHY와 마찬가지로 RRC가 가지는 역할을 유연하게 수행할 수 있도록 시나리오를 기반으로 FSM 구조로 RRC 프로토콜의 구조를 표현하였다. Cell search 시나리오에 따라 기본적으로 RRC에서 가져야 하는 상태나 발생 이벤트를 기반으로 설계되었다. Cell search 관점에서 RRC는 serving cell이 항상 존재하는 상황을 추구하며 serving cell이 없는 경우에는 지속적으로 cell search를 시도하여 serving cell이 될 수 있는 cell을

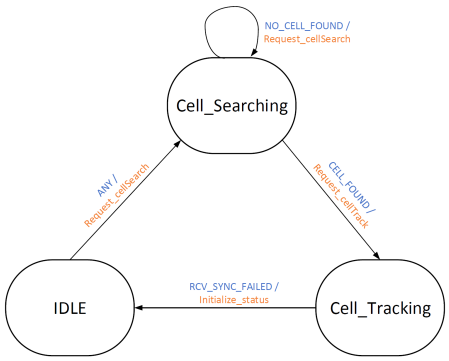

그림 9. RRC 과정에 대한 FSM
Fig. 9. RRC Process FSM

궁극적으로 찾아야 한다.

초기 상태에 해당하는 IDLE 상태에서 RRC는 곧바로 cell search를 위한 service primitive를 PHY로 전달한다. 이때 주파수는 마지막으로 머무른 cell의 주파수로 지정하거나 단말에서 지원 가능한 RF channel 중 한가지를 임의로 선택한다. RRC FSM은 PHY에게 cell search service primitive를 전달하면서 Cell_Search 상태로 천이된다. 그리고 이 상태에서 PHY로부터 cell search에 대한 결과를 얻으면 cell search 성공 여부에 따라 다른 후속 동작을 취한다. Cell search에 실패한 경우 그 다음 순서에 해당하는 주파수를 지정해서 cell search를 다시 시도한다. 만약 PHY에서 전달받은 cell search 결과가 성공이고 관련 cell 정보를 획득하면 RRC는 S-criterion 기준을 따지기 위해 cell tracking을 위한 service primitive를 전달하고 이때 RRC FSM은 Cell_Tracking 상태로 천이된다. 이 Cell_Tracking 상태에서 PHY는 이전에 탐색한 cell에 대해 동기화를 지속적으로 유지하면서 shared channel을 수신하게 된다.

Cell_Tracking 상태에서 만약 PHY가 cell search 혹은 동기화 유지에 실패하면 RRC는 다시 IDLE 상태로 돌아가서 cell search를 처음부터 반복하게 된다. 이와 같은 상태 회귀는 단말이 cell을 놓치는 시나리오에서 RRC로 하여금 새로운 cell을 찾기 위한 cell search 과정의 반복을 손쉽게 구현할 수 있게 한다.

Ⅳ. 코드 구조 설계 및 구현

본 장에서는 앞에서의 설계를 기반으로 실제 코드 구현하는 과정을 설명하고 그 구현 결과를 OAI의 코드 와 비교분석하고자 한다. OAI 코드는 그림 10과 같은 구조를 가지며, 코드 구조의 특성 상 한번의

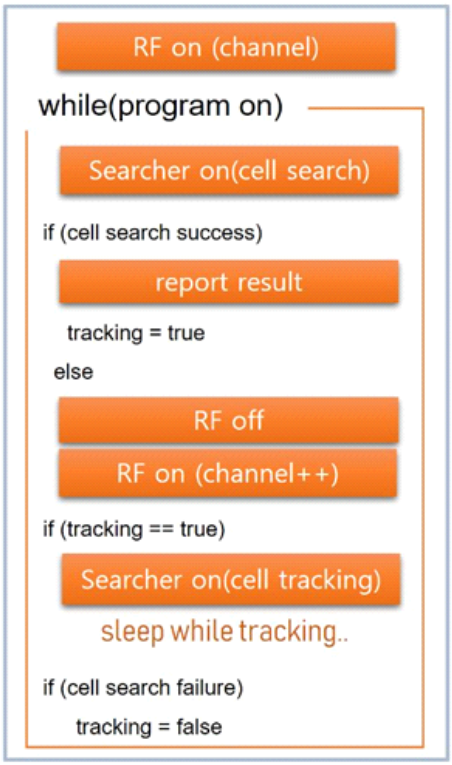

그림 10. OAI 코드의 동기화 과정
Fig. 10. OAI synchronization process

thread 동작에 한번의 cell search과정을 직렬로 수행하는 형태로 구현되어 있다. 한번의 while loop 반복을 통해 지정된 RF channel에 대해 cell search가 한번 수행되며 여러번의 while loop 반복을 통해 여러 RF channel에 대한 순차적 cell search를 수행할 수 있다. 이 과정에서 cell이 발견하면 해당 cell에 대해 지속적인 동기를 유지(tracking)하는 handler 코드로 넘어간다. 그리고 cell search 및 tracking 실패 시에는 RF 등의 초기화와 함께 다른 RF channel로 지정하여 cell search를 다시 시도하게 된다.

위의 코드 구조에서는 관련 동작들이 한 코드 영역(while문)에 평면적으로 구현되어 있다. 이런 구현 구조에서는 특정 코드 영역에 복합적인 동작이 기술된 형태이므로 가독성이 떨어지고 개발 과정에서 코드의 흐름을 따라가는데 많은 시간을 소모하게 한다. 또한 이런 코드 구조에 새로운 cell search 시나리오를 적용하는 경우 기존 코드를 재활용하기 어렵고 새로운 코드를 추가하는 형태로 구현되어 코드가 더 길고 비효율적인 형태가 될 가능성이 크다. 기본적으로 cell search의 알고리즘 및 각 계층 프로토콜은 분리된 구조로 구현이 이루어져야 표준과의 호환 유지 측면에서 바람직하다. 그러나 현재 OAI의 평면적 코드 구조는 그렇지 않다. 가령 OAI 코드에서 PHY 계층 관련 코드에서 시나리오 흐름에 따라 RF channel을 자체적으로 조정하게 구현되어 있으나, 이는 표준 상에서는 RRC 계층의 역할이다. 이런 구조에서는 RF channel을 선택하는 시나리오를 유연하게 수정 구현하기 어렵다.

이에 반해 본 연구를 통해 구현한 FSM 기반 설계의 코드는 그림 11과 같은 구조를 가진다. 상위 계층에서 cell search에 대한 service primitive를 받으면 PHY는 메시지를 scheduler queue에 넣은 뒤, 순차적으로 cell search를 수행한다. Scheduler는 설계된 FSM에 포함된state 및 event에 따라 지정된 action을 수행하는 형태로 cell search 과정을 수행한다. 이 action은 각 state 천이에 따라 handler 함수 형태로 구현하고 state 천이 유형에 따라 적절한 handler 함수로 mapping이 되어 호출되도록 구현하였다. Scheduler에서 생성된 FSM은 RRC로부터의 service primitive 별로 생성이 되며, IDLE 혹은 RF_sync state에서 시작하여 다시 원래의 state로 되돌아오면서 1회 cell search 신호처리 과정이 완료된다. 특정 FSM은 적절한 event가 발생할 때까지 현재 state를 유지하며, event 발생에 따라 그에 대응되는 다음 state를 산출하고, state 천이에 맞는 handler 함수를 호출한다.

그림 12는 앞의 코드 구조를 기반으로 실제 구현된 scheduler와 FSM 코드이다. State 천이에 맞게 handler 함수를 가리키는 함수포인터를 행렬 형태로 정의하고 현재 state와 event에 따라 적절한 handler

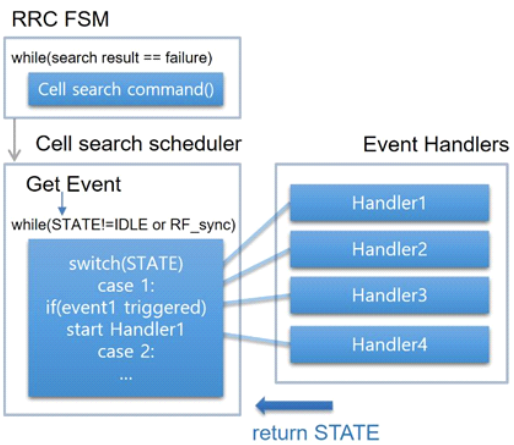

그림 11. 제안된 동기화 과정의 흐름도
Fig. 11. Proposed synchronization process

그림 12. Cell Search Scheduler에 대한 FSM 구현
Fig. 12. Cell Search Scheduler FSM code

함수를 호출하도록 다양한 시나리오에도 일관된 코드 처리로 구현하였다.

여기서 구현된 코드는 RRC와 PHY의 역할이 표준 내용에 맞게 명확하게 구분되어 구현될 수 있으며, PHY 관련 thread의 계산 부담이 줄어드는 장점을 가진다. 또한 state 천이에 따라 구조적으로 해당 시나리오를 동작시킬 수 있으며, handler 함수 기반의 모듈화된 구현으로 인해 중복된 코드 구현을 방지하여 코드의 재활용성을 높일 수 있다. 뿐만 아니라, 새로운 시나리오를 적용하고자 할 때 새로운 시나리오 상황에 맞는 state, event 및 handler 추가 정의를 통해 직관적이면서 용이한 수정 개발이 가능하다.

V. 실험 및 결과 분석

구현된 프로토콜의 동작을 확인하기 위해 본 연구에서 가정한 실험 환경은 그림 13과 같다. 리눅스 기반 PC에 본 연구를 통해 구현한 시스템 코드를 구축하고 연결된 USRP와 연동하여 소프트웨어 모뎀이 실제 상용 신호 송수신을 수행하도록 하였다. 실제 실시간으로 동작하는 리눅스 기반 PC에서 RF 및 Searcher 그리고 제안된 구조의 프로토콜에 대한 소프트웨어 구현을 구동시키는 형태로 실험하였다. Table 1은 본 연구에서 활용한 PC의 하드웨어 사양을 나타낸다. USRP에 상용 5G 주파수 대역에 해당되는 3.5GHz 대역의 안테나를 연동하여 실제 5G 기지국의 신호를 수신하는 상황에서 cell search를 수행하는 실험을 하였다.

표 1. 테스트베드의 하드웨어 구성
Table 1. H/W Specification

CPU	Intel core i7-8700k CPU @ 3.70GHz x6
RAM	16G
OS	Ubuntu 16.04 LTS
SDR	Ettus USRP B210 70MHz-6GHz

Table 2는 본 실험 환경과 관련된 frame configuration parameter이며 Max error count는 한번 cell search 명령을 수신하였을 때 PHY에서 반복하는 cell search 신호처리의 최대 오류 횟수를 말한다. 즉, cell search 신호처리 관점에서 3회 실패하면 최종적으로 해당 RF channel에 대한 cell search 과정이 실패한 것으로 판단한다고 볼 수 있다.

본 연구에서 제안한 코드 구조의 효율성을 확인하기 위하여 동작 중인 cell search 알고리즘을 임의의 시점에 중단시키는 새로운 시나리오를 현 코드에 적용해 보았다. 그림 14와 같이 scheduler에 의해 동작되던 모든 cell search 과정을 도중에 취소하여 thread

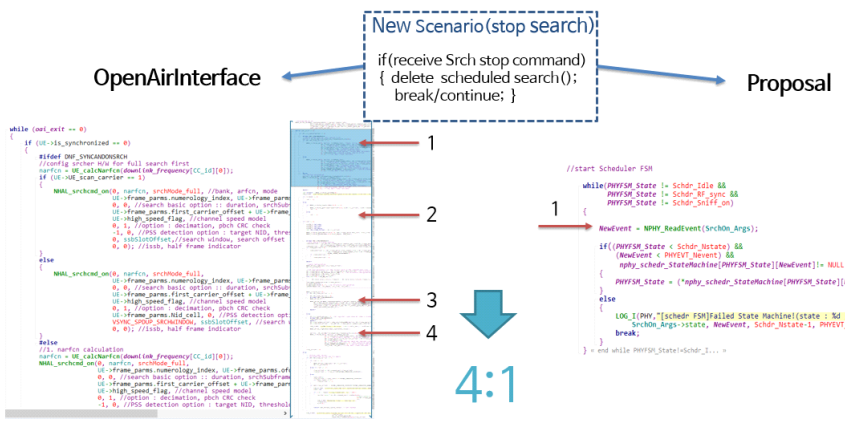

그림 13. 새로운 시나리오 구현 과정에서의 효율성 비교
Fig. 13. Comparison of Implementational Efficiency for Adding a New Scenario

표 2. Frame 구성 파라미터
Table 2. Frame Configuration Parameters

Parameters	Value
Band	78
ARFCN	640586
Cell ID	923
Bandwidth	20MHz
Sampling rate	30720000
Subcarrier spacing	30kHz
FFT size	1024
Max error count	3

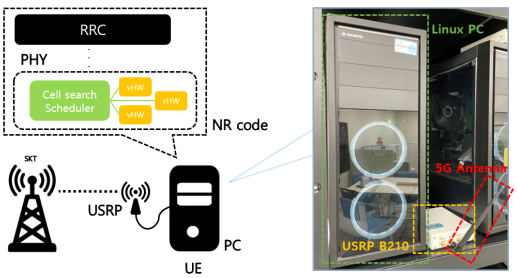

그림 14. 테스트베드 구성
Fig. 14. Testbed configuration

를 종료하는 구현이 추가되어야 한다. 본 연구에서 구현한 코드의 경우 FSM 코드의 한 부분에 과정을 종료하는 코드를 추가하면 각 state에서 새로운 시나리오가 실현되도록 구현될 수 있다. 반면 기존 OAI 코드 구조에서 동일 효과를 내기 위해서는 평면적으로 cell search 신호 처리 및 시나리오 처리를 수행하는 thread 코드 구조 하에서 동일한 코드를 최대 4회 추가되어야 구현이 가능하다.

실제 본 연구의 구현 코드에 위와 같은 시나리오를 구현하기 위해 stop search라는 event를 새로 추가하고 FSM 코드 상단에 stop search event에 대한 action 코드를 추가하였다. 그리고 이때 반복 실험을 통해 동작 처리 시간을 측정하였다. 각 cell search 시도마다 cell search 과정에 걸리는 시간을 측정하고, 100회 수준의 반복을 통해 소요시간의 추이를 살펴보았다.

그 실험 결과는 그림 15와 같다. OAI 코드의 경우 모든 cell search 과정이 끝날 때 stop search event 발생 여부를 확인하므로 매번 시험 시도마다 거의 일정한 소요시간이 측정되었다. 반면 제안된 코드 구조의 경우 state의 천이마다 stop search event의 발생 여부를 확인하므로, event 발생 타이밍에 따라 소요시간이 절약되는 것을 볼 수 있다. 평균 관점에서 반복 실험

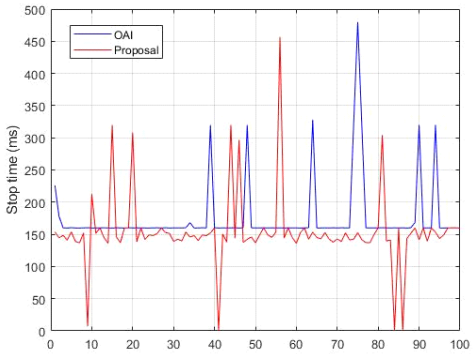

그림 15. 'Stop search' 시나리오 적용 후 멈추기까지의 소요시간 실험 결과
Fig. 15. Time Consumption of 'Stop Search' Scenario

결과를 비교할 때, OAI 코드의 평균 소요시간은 161ms 수준인 반면, 제안 코드 구조는 142ms로 약 12% 소요시간을 절약할 수 있는 것을 확인할 수 있었다. 본 연구에서 구현한 코드는 구조 측면에서 stop search event 발생 시 전체 cell search 과정을 수행하지 않는 구현을 가능하게 한다. 이는 본 연구에서 제안한 코드 구조가 기존 OAI 대비 더 유연하고 효율적인 시나리오 구현을 가능하게 함을 시사한다고 볼 수 있다.

VI. 실험 및 결과 분석

본 연구에서는 빠르게 발전하는 이동통신 시장에 맞추어 유연하게 대처할 수 있는 소프트웨어 모뎀의 프로토콜 구조를 제안하였다. 최신 이동통신에 해당하는 5G 시스템의 cell search 과정을 유연하고 효율적으로 구현할 수 있는 코드 구조를 설계하였다. 코드의 가독성과 개발 유연성을 높이기 위해 프로토콜의 소프트웨어 구조를 FSM 형태로 설계하고 제안된 코드 구조의 우수성을 증명을 위해 OAI 코드 구조와 비교하였다. 그리고 'stop search'관련 새로운 시나리오를 구현할 때, 제안된 코드 구조가 OAI 코드에 비해 소요시간이 적고 효율적인 구현이 가능함을 확인할 수 있었다.

본 연구에서는 빠르게 발전하는 이동통신 시장에 맞추어 유연하게 대처할 수 있는 소프트웨어 모뎀의 프로토콜 구조를 제안하였다. 최신 이동통신에 해당하는 5G 시스템의 cell search 과정을 유연하고 효율적으로 구현할 수 있는 코드 구조를 설계하였다. 코드의 가독성과 개발 유연성을 높이기 위해 프로토콜의 소

프트웨어 구조를 FSM 형태로 설계하고 제안된 코드 구조의 우수성을 증명을 위해 OAI 코드 구조와 비교하였다. 그리고 'stop search'관련 새로운 시나리오를 구현할 때, 제안된 코드 구조가 OAI 코드에 비해 소요시간이 적고 효율적인 구현이 가능함을 확인할 수 있었다.

References

[1] T. Ulversoy, "Software defined radio: Challenges and opportunities," *IEEE Commun. Surv. & Tuts.*, vol. 12, no. 4, pp. 531-550, May 2010.

[2] N. B. Truong, Y. Suh, and C. Yu, "Latency analysis in GNU Radio/USRP-Based software radio platforms," *IEEE Military Commun. Conf.*, pp. 305-310, San Diego, CA, USA, Nov. 2013.

[3] P. Lin and S. Huang, "Performance profiling of cloud radio access networks using OpenAirInterface," *2018 APSIPA ASC*, pp. 454-458, Honolulu, HI, USA, Nov. 2018.

[4] A. Virdis, N. Iardella, G. Stea, and D. Sabella, "Performance analysis of OpenAirInterface system emulation," *3rd Int. Conf. Future Internet of Things and Cloud*, pp. 662-669, Aug. 2015.

[5] F. Kaltenberger, G. d. Souza, R. Knopp, and H. Wang, "The OpenAirInterface 5G New radio implementation: Current status and roadmap," *23rd Int. ITG Wkshp. Smart Antennas*, pp. 1-5, Apr. 2019.

[6] S. Huang, Y. Su, Y. He, and S. Tang, "Joint time and frequency offset estimation in LTE downlink," *7th Int. Conf. Commun. and Netw. in China*, pp. 394-398, Aug. 2012.

[7] 3GPP TS 38.304, v15.4.0, *NR; User Equipment (UE) procedures in Idle mode and RRC*, Jun. 2019.

[8] 3GPP TS 38.101-1, v16.3.0, *NR; User Equipment (UE) radio transmission and reception*, Jun. 2020.

[9] 3GPP TS 38.211, v15.4.0, *NR; Physical channels and modulation*, Dec. 2018.

[10] 3GPP TS 38.213, v15.4.0, *NR; Physical layer procedures for control*, Dec. 2018..

[11] 3GPP, TS 38.214, v15.4.0, *NR; Physical layer procedures for data*, Dec. 2018.

[12] 3GPP TS 38.331, v15.4.0, *NR; Radio Resource Control (RRC); Protocol specification*, Dec. 2018.

[13] 3GPP TS 38.133, v15.4.0, *NR; Requirements for Support of Radio Resource Management*, Dec. 2018.

김 혜 원 (Hyewon Kim)

2017년 2월 : 한양사이버대학교 해킹보안과 학사

2021년 2월 : 숙명여자대학교 전자공학과 석사

2021년 6월~현재 : 콘텔라 전임연구원

<관심분야> 이동통신, 소프트웨어 모뎀, Access network

김 주 엽 (Juyeop Kim)

2010년 4월 : KAIST 전자전산학과 학사

2010년 1월 : KAIST 전기 및 전자공학과 박사

2011년 4월~2013년 12월 : 삼성전자 무선사업부 (책임연구원)

2014년 1월~2018년 2월 : 한국철도기술연구원 (선임연구원)

2018년 3월~현재 : 숙명여자대학교 전자공학전공 조교수

<관심분야> 이동통신, 소프트웨어 모뎀

[ORCID:0000-0003-4262-6063]