

객체지향프로그램을 이용한 CDMA 계층 셀 시뮬레이터 개발

김 호 준

전주대학교 전기전자정보통신공학부

Development of CDMA Hierarchical Cellular Simulator using Object-Oriented-Program

Ho-Joon Kim

School of Electrical, Electronic, Information and Communication Engineering

This paper presents design and development of a simulator evaluates the performance of a hierarchical cellular system. The proposed hierarchical cellular simulator, consisting of macro, micro, and pico cells, applies the wrap-around technique to reduce simulation time. The simulator is implemented as object oriented class models by using the C++ language in a PC environment. The resulting application can evaluate the interference, SIR(Signal to Interference Ratio), and capacity of a hierarchical cellular system in various configurations. Moreover, it can be used in other applications such as power control, call admission control, hand over scheme.

Keywords : CDMA, Hierarchical Cellular, Simulation, Object Oriented Program

1. 서 론

장소에 제한 없이 이동 중에도 통화를 할 수 있는 장점으로 인해 셀룰라 이동 통신 가입자는 지난 20여 년간 엄청난 증가가 있었다. 이런 고 성장은 시장 규모를 증가 시켰고 셀룰라 망 사업자는 계속되는 수요 증가에 대응을 해야 했다. 특히 사용자는 더 빠른 속도 더 좋은 품질을 지속적으로 요구해왔다. 즉, 사용자는 쇼핑센터, 기차역, 공항 등과 같은 공공장소를 포함한 어느 곳에서나 고품질의 통화를 원하고 통화 차단 확률이나 통화 단절 확률이 낮아지기를 기대하고 있다. 셀룰라 시스템은 유한한 전파자원 제약 하에서 용량 증대와 통화 품질 향상을 위해 주파수 재사용과 셀 분할 방식이 적용되었고 다중접속방식이 FDMA에서 TDMA, 그리고 CDMA로 진화가 되었다[10].

셀룰라 시스템은 셀 분할과 색터화를 통해 시스템 용량이 증대되었으나 이는 기존의 매크로 셀 구조를 근간으로 하는 시도였다[11]. 매크로 셀은 고출력 기지국이

사용되고 직접 전파가 도달하지 못하는 음영 지역 같은 곳에서는 통화가 불가능하거나 시스템 용량 효율에 장애가 되었다. 셀 크기가 작은 마이크로 셀이나 피코 셀로 갈수록 직접파 도달 확률이 높아지므로 저출력 기지국을 사용해도 되는 이점이 있어 1Km 반경 이내의 소규모 셀의 적용이 많아지고 있다. 마이크로 셀 사용자는 길모퉁이에서 급격한 신호 감쇄를 겪으며 고속 단말은 셀 경계를 넘을 때 핸드오버를 처리해줄 시간 적 여유가 적기 때문에 통화 단절을 격기도 한다. 또한 잦은 핸드오버는 시스템의 시그널링 부하를 증대 시킨다. 따라서 고속 이동 단말은 매크로 셀이 담당함으로써 핸드오버 빈도를 줄여줄 필요가 있다. 이에 따라 기존의 셀룰라 시스템 개념을 확장한 계층 셀룰라 개념이 필요하다. 국부적인 통화 밀집 지역(hot spot)은 마이크로 또는 그 보다 작은 규모의 셀이 담당하고 넓은 지역은 연속적인 서비스 제공을 위해 매크로 셀이 담당하는 것이 유리하다. 그리고 인구 밀집지역에서는 매크로와 마이크로 셀 모두가 서비스함으로써 연속적인 서비스가 가

능하다[7].

3세대 시스템은 실내에는 피코 셀을, 도심 지역에는 마이크로 셀 또는 매크로 셀을 제공하였다. 더욱이 위성을 이용해 광역 커버리지를 제공하여 지상국에서 담당하지 못하는 영역은 메가 셀이 담당하게 할 수도 있다[12]. 미래의 셀룰라 통신망은 다양한 셀 규모가 공존하여 각기 다른 속성의 서비스들이 최적의 셀에 할당되어 서비스되는 구조로 운영되어야 한다[3, 4]. 이러한 계층 구조 셀룰라 시스템의 특성을 분석하는 데는 수학적 모델링에 의한 해석으로는 다양한 변수를 적용하는데 한계가 있다. 따라서 계층 셀 시스템의 링크 성능 분석, 용량 평가, 자원관리, 시스템 운영 등의 연구를 위해 시뮬레이션 도구가 필요하다. 상용 시뮬레이터로 NS-2, OPNET 등이 널리 사용되나 이는 LAN, IS-95, GSM 등과 같이 확정된 구조의 셀룰라 시스템에서 파라미터 변화에 따른 성능 분석에 유용하다. 그러나 새로운 계층 셀과 같은 새로운 구조의 시스템을 검증하기 위해서는 전용 시뮬레이터가 필요하다. 지금까지 시뮬레이션을 이용하는 계층 셀 시스템 연구는 있었으나 셀 구성이 고정된 형태여서 실지 시스템을 정확히 반영하지는 못하였다[8, 9]. 따라서 매크로, 마이크로, 피코 셀 모두가 존재하며 자유로이 셀 규모를 확장할 수 있는 계층 셀 시뮬레이터를 개발할 필요가 있다.

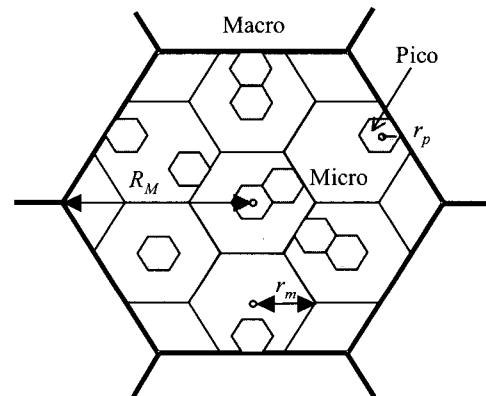
일반적으로 시뮬레이션 속도와 코드 개발의 편의성을 위해 C 프로그램을 사용하는 경향이거나 시스템 규모가 커지고 기능이 다양화 되면 개발의 편의성, 이식성, 재사용성을 고려하면 객체지향 프로그래밍 기법의 시뮬레이터가 효율적이다. 따라서 본 연구에서는 계층 셀 시뮬레이터를 개발하기 위한 셀 모델과 객체 설계, 시뮬레이션 처리 알고리즘을 연구 개발하고 개발된 시뮬레이터를 사용한 시뮬레이션 응용 예를 제시하고자 한다.

2. 계층 셀 모델

본 논문에서의 계층 셀은 매크로 셀, 마이크로 셀, 피코 셀의 세 계층을 고려하며 매크로 셀은 반경 10Km 내외의 크기이고, 마이크로 셀은 반경 1Km 내외의 크기이며, 피코 셀은 반경 200m 내외로 한다. 매크로 셀에 마이크로 셀이 중첩되어 공존하며 마이크로 셀에 피코 셀이 중첩하여 공존한다. 매크로 셀은 커버리지 공백이 없이 완전하게 분포하고 마이크로 셀과 피코 셀은 일부 셀만 존재하는 불완전한 셀 구성 모델이 가능하도록 한다. 매크로 셀과 마이크로 셀 및 피코 셀의 반경은 각각 R_M , r_m , r_p 로 정의하며 각 계층 셀 반경의 비율 $\mu_m = R_M / r_m$, $\mu_p = r_m / r_p$ 로 정의하고 비율에 따라 각 계

층 셀의 상대 크기와 존재 가능한 최대 하위 계층 셀의 개수를 결정한다. 피코 셀의 경우 수직으로 N_p 층이 존재하는 건물로 모델화한다. 정육각형 셀 구조의 특성에 따라 완벽히 분포하는 k 겹(tiers) 셀 군은 $6k$ 개의 셀이 존재하나 셀 분포율 P 를 정의하여 $6kP$ 개의 셀만 존재하는 구성을 모델화한다. 이때 $0 \leq P \leq 1$ 의 값을 갖는다.

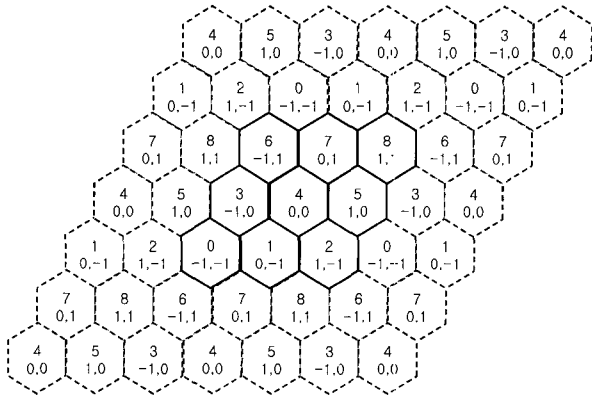
반경비율 비율 μ_m 과 μ_p , 그리고 마이크로 및 피코 셀 분포율 P_m 과 P_p 에 따라 구성된 각 계층 셀의 개수를 살펴보면 다음과 같다. 하나의 매크로 셀에는 $P_m \mu_m^2$ 개의 마이크로 셀이 존재하며 하나의 마이크로 셀에는 $P_p N_p \mu_p^2$ 개의 피코 셀이 존재한다. <그림 1>은 $\mu_m = \mu_p = 3$ 이고, $P_m = 1$, $P_p = 1/6$ 이며, 피코 셀은 10층 건물의 각 층에 해당할 경우, 하나의 매크로 셀에 9개의 마이크로 셀이 그리고 피코 셀은 k 겹(tier) 마다 kN_p 개의 셀이 존재하여 한 마이크로 셀에는 15개의 피코 셀이 존재하는 모델 형태이다.



<그림 1> 계층 셀 모델

셀룰라 시스템을 모델링 할 때는 실지 시스템에서 얻을 수 있는 결과와 가장 유사한 결과를 얻도록 하는 것이 최대 목표가 된다. 실지 시스템은 수천 개의 셀로 구성되어 있다. 물론 이와 똑같이 모델링 할 수도 있겠지만 모든 셀을 시뮬레이션 한다는 것은 엄청난 시간 소요를 필요로 하는 난점이 있다. 따라서 이 연산 시간을 줄이기 위해서는 실지 시스템을 대표할 수 있는 적은 수의 셀만을 선택하여 시뮬레이션 할 수 밖에 없다. 단, 시뮬레이션 결과가 실지 시스템에서 얻을 수 있는 결과에 상응할 만한 셀 수를 선택해야 한다. 실지 시스템에서는 모든 셀에 주변 셀들이 싸여 있으나 유한 셀 시뮬레이션 모델에서는 정 중앙의 셀만 실지 환경과 유사하고 외곽의 셀들은 충분한 주변 셀이 존재하지 않는 상태가 된다(edge effect). 이를 해결하기 위해 측정 대상인 중심 셀을 먼저 구성하고 측정을 앓을 주변 셀들을 다수 배치하는 방법이 있다. 이때 주변 셀들은 측정이

필요하지는 않으나 중심 셀과 동일한 처리는 해 주어야 한다. 따라서 모델링 한 셀의 수가 많아지면 시뮬레이션 시간이 너무 길어진다. 이런 문제점을 개선하기 위한 다른 방법으로 마치 모든 셀이 주변 셀의 중심에 있는 것 같은 효과를 내는 wrap-around 기술이 있다[2, 4]. 연구 대상 셀을 마름모형으로 배열하고 위쪽 끝 셀들이 아래쪽 끝 셀들과 맞닿도록 하고 왼쪽 끝 셀들을 오른쪽 끝 셀들과 맞닿도록 하면 도우넛형 또는 타이어 튜브형 입체 표면에 셀들이 존재하는 형태가 된다. 이는 모든 셀이 모든 방향으로 주변 셀을 갖고 있는 것처럼 여길 수 있다. 이렇게 하면 유한하면서 경계가 없는 실지 셀룰라 시스템처럼 시뮬레이션 모델을 설정할 수 있다. 이를 wrap-around 셀 모델이라 하며 중심 셀을 기준으로 평면에 펼쳐 놓은 형태가 <그림 2>이다.



<그림 2> 펼친 torus 모양의 셀 구성

wrap-around 시뮬레이션 기법은 외곽의 셀은 중심 셀에 비해 간섭이 낮게 평가되는 edge-effect가 없으며 모든 셀의 측정값을 다 사용할 수 있어 통계의 신뢰도가 높아지고 따라서 시뮬레이션 시간을 줄일 수 있는 장점이 있다. 지금까지 이 기법을 사용한 분석이 있었으나 셀 개수가 고정되었고 실지 시스템과 유사하도록 확장함으로써 모델의 차이에 따른 분석의 정확도를 비교할 수는 없었다[5, 9, 13]. 위와 같은 wrap-around 기법을 이용함으로써 본 연구의 시뮬레이터는 계층 셀 기본 구성 외곽에 한 겹의 주변 셀들이 존재하도록 설정하여도 여러 겹의 셀들이 존재하는 실지 환경과 같은 결과가 나오면서 시뮬레이션 시간이 1/100 만큼 감소되도록 동작한다[2].

일반적으로 셀룰라 시스템 시뮬레이션을 할 때 다음 두 가지 방법이 사용된다. 첫째는 일정한 시간 간격을 두고 새 호의 시작, 호의 종료, 단말 위치 변경을 일괄 처리하는 discrete time step 방법이고 둘째는 새로운 호의 시작, 호의 종료 또는 단말의 위치 변경 등의 상황

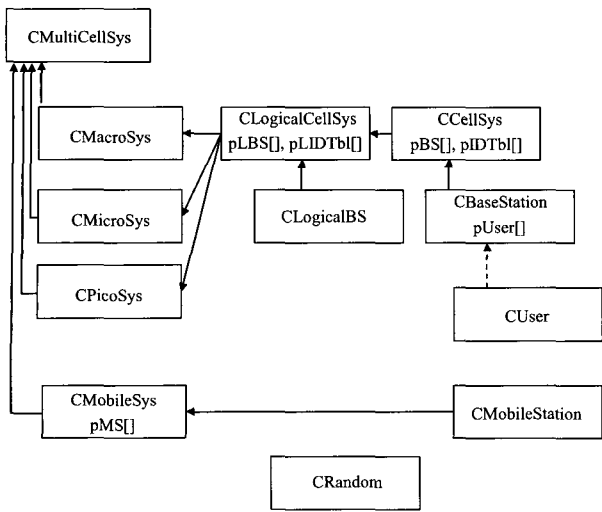
이 바뀔 때마다 처리를 하는 event driven 방식이다[6]. 전자는 snap-shot 방식이라고도 하며 한 순간에 전체 시스템의 동작을 일괄 처리할 수 있으며 기존의 연산 소프트웨어를 활용하기 용이하고 병렬처리를 통해 시뮬레이션 시간을 단축하기에 적당한 장점이 있다. 후자는 real-time 방식이라고도 하며 이벤트가 발생한 시각에 정확하게 시스템 동작을 처리하기 쉽고 이벤트를 놓칠 우려가 없으며 이벤트 발생이 없을 경우 불필요한 연산을 할 필요가 없는 장점이 있다. discrete time step 방법이 상대적으로 수행 속도 면에서 유리하다.

본 연구의 시뮬레이터는 신호대 간섭비, 시스템 용량, 부하 분포 등 정적인 성능 지표를 빠른 시간에 분석하기 위해서는 discrete time step 방식을 사용하고 트래픽량의 변화, 핸드오프의 처리 등 동적이고 시간 변화적 처리의 성능을 분석하기 위해서는 event driven model 방식을 사용도록 선택할 수 있게 되어있다.

3. 시뮬레이터 설계

3.1 클래스의 구성

시스템의 구성 요소인 기지국과 단말기 그리고 다수의 기지국으로 구성된 매크로, 마이크로, 피코 시스템, 다수의 단말로 구성된 단말 시스템, 그리고 전체 프로그램의 운영을 통합 관장하는 멀티 셀 시스템 등이 중요 구성 요소이다. 기타 랜덤 수 발생기 및 그래픽 유저 인터페이스를 제공하는 메뉴 및 팝업 다이얼로그 운영 프로그램 등이 있다. 각 클래스의 상호 파생 관계를 표현한 것이 <그림 3>이다. CBaseStation 클래스는 기지국의 파라미터, 접속된 사용자의 데이터, 그리고 동작 멤버 함수를 갖고 있다. CCellSys 클래스는 설정된 계층 수만큼의 실존 기지국의 데이터를 갖고 동적 메모리 할당으로 프로그램 시행 시 메모리 용량을 최소화 할 수 있게 되어있다. CLogicalCellSys 클래스는 wrap-around 기법에 따른 가상 기지국의 데이터를 갖고 있고 실존 기지국과의 대응을 위한 테이블을 갖고 있다. 이 CLogicalCellSys 클래스는 매크로 셀, 마이크로 셀, 피코 셀 구성을 위해 각각 CMacroSys, CMicroSys, CPicoSys 클래스에서 사용된다. CMobileStation 클래스는 단말기의 속성과 동작 멤버 함수를 갖고 있으며 CMobileSys 클래스에서 사용자 수만큼 동적 데이터 할당을 한다. CMultiCellSys 클래스는 이 모든 클래스로 구성되며 각종 파라미터의 설정과 목표 시뮬레이션 멤버 함수를 갖고 있다. CRandom 클래스는 각종 분포 함수 발생을 위한 멤버 함수들로 구성되어 있다.



<그림 3> 시뮬레이터의 주요 클래스 구성도

<그림 4>는 시뮬레이터 클래스 설계의 한 예로서 최상위 클래스인 CMultiCellSys 클래스를 간략화 한 프로그램이다. 프로그램은 public 변수 객체를 최소화하고 public 멤버 함수를 통해 변수를 access하도록 하는 객체 지향 프로그래밍 개념에 충실하게 함으로써 안정성과 이식성이 좋도록 설계하였다.

```

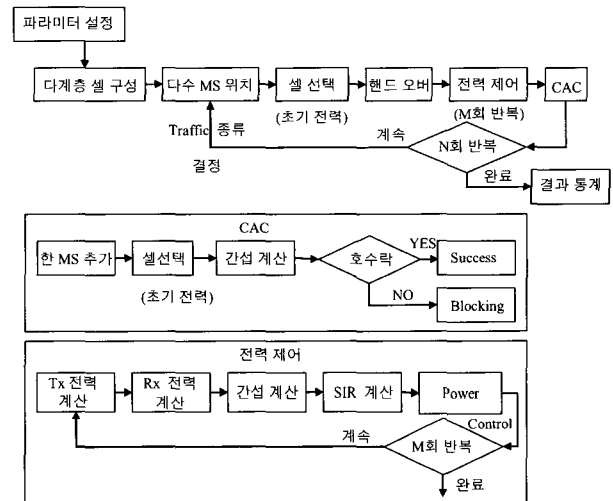
class CMultiCellSys
{public:
    CMultiCellSys(); // 생성자
    virtual ~CMultiCellSys(); // 소멸자
    CMacroSys MacroSys; // 매크로셀 오브젝트
    CMicroSys MicroSys; // 마이크로셀 오브젝트
    CPicoSys PicoSys; // 피코셀 오브젝트
    CMobileSys MobileSys; // 단말시스템 오브젝트
    SetupMultiCellSys(void); //계층셀형성(메모리확보)
    InitMultiCellSys(); // 계층셀 초기화(변수 초기화)
    int GetNumofSector(void) const
private:
    int NumofFA ; // 주파수 캐리어 수
    int NumofLayer ; // 계층 수
    int NumofSector ; // 섹터 수
    int MaxPctrlNum ; // 전력 제어 횟수
    ... };
    
```

<그림 4> 클래스 설계의 예(셀 시스템 구성)

3.2 Snap Shot 시뮬레이션

이는 임의의 시스템 상태를 랜덤하게 설정하고 그 상태에서의 성능 지표를 얻어내고 이러한 과정을 여러 번

실행하여 평균을 내는 확률 통계적 방식으로 <그림 5>에 흐름도를 나타내었다. 파라미터 설정 단계에서는 계층의 개수, 각 계층의 기지국 수, 반경, 파일럿 신호 세기, 물리 셀과 논리 셀의 개수, 전파 환경 정보, 반복 시행 횟수 등의 시스템 구조와 시뮬레이션 조건 등을 설정하는 단계이다. 다 계층 셀 구성 단계에서는 시뮬레이션 조건에서 정한 물리 셀 및 논리 셀의 개수만큼 메모리를 할당하고 매크로, 마이크로, 그리고 피코 셀을 구성하여 기지국의 위치, 기지국의 종류, 셀의 최대 수용 사용자수 등의 초기 파라미터를 설정한다. 또한 시스템의 최대 수용 단말 개수만큼 메모리를 할당하고 단말의 기본 정보를 초기화한다. 다수 MS 위치 단계에서는 트래픽 종류에 따라 단말의 위치를 무작위로 결정하며 셀 선택 단계에서 현 위치에서 가장 신호 세기가 강한 기지국을 찾고 셀 선택 알고리즘에 따라 매크로, 마이크로, 피코 셀로의 접속을 시도한다. 또한 단말의 이동 또는 채널 환경의 변화에 따라 핸드오버와 전력제어를 실시하며 호 수락 제어(CAC ; Call Admission Control) 단계에서 새로 접속을 시도한 단말을 접속시킬 것인지 선택한 계층의 기지국에서 결정한다. 다 계층 셀 구성 단계 이후는 N회 반복되며 N회 동안 시스템에서 나타난 각종 파라미터의 값을 통계 낸다.



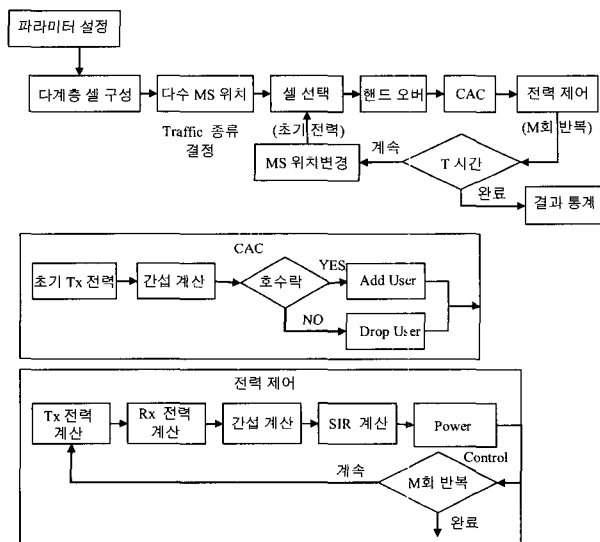
<그림 5> Snap shot 시뮬레이터의 처리도

전력 제어는 단말 또는 기지국에서의 초기 송출 전력을 결정하여 모든 셀에서의 간섭량을 계산하고 신호대 간섭비를 계산하며 목표 신호대 간섭비 임계값과 비교하여 전력의 증가 또는 감소를 처리한다. 이 과정을 M회 반복하여 안정 상태가 되도록 한다. 호 수락 제어는 먼저 하나의 단말을 추가하여 셀 선택을 하고 간섭을 계산한 후 신호대 간섭비를 계산하고 호 수락 제어 알

고리침에 따라 호 수락 여부를 결정한다. 호 수락일 경우와 실패의 경우를 누계 한다.

3.3 Real Time 시뮬레이션

이는 시간 진행에 따라 시스템 각 구성 요소의 동작을 구현하여 장시간 동안의 단말의 접속 시도와 접속해지, 그리고 단말의 이동과 전파 환경 변화에 따른 핸드오버 처리를 이벤트 발생 시점마다 하면서 특정 파라미터를 통계 내는 방식이며 <그림 6>에 그 흐름도를 나타내었다. 파라미터 설정 단계에서 정해진 시스템 부하(Offered Traffic)에 상응하는 수 만큼의 단말을 배치하고 각 단말의 호 유지 시간에 따라 호 해지와 신규 호 발생이 시간의 흐름에 따라 진행된다. CAC 단계에서는 단말의 초기 송신 전력을 결정하고 새로운 호가 추가되었을 때의 간섭량을 다시 계산하여 허용 SIR이 유지될 수 있으면 호를 수락하고 그렇지 않으면 드롭 처리를 하여 통계를 갱신한다. 전력 제어 단계는 단말의 이동에 따라 변화된 SIR에 맞추어 모든 단말의 송신 전력을 정해진 스텝 만큼 가감한다. 또한 변화된 SIR 값을 다시 계산하여 CAC 단계에서 호 수락의 기준으로 사용하게 한다. 이 과정은 호 수락 제어 알고리즘에 따라 다른 함수를 function pointer 방식으로 선택 실행할 수 있다.



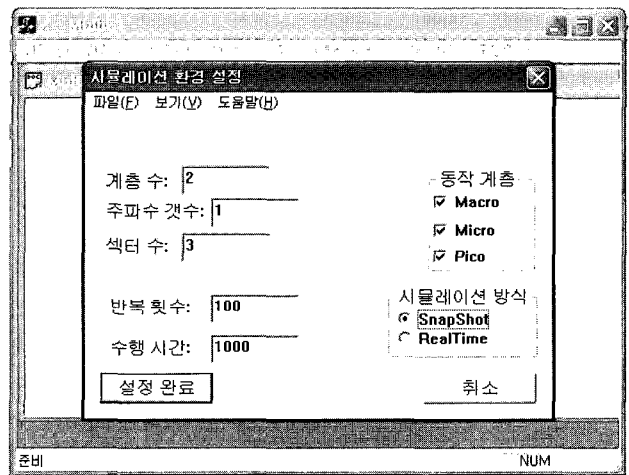
<그림 6> Real Time 시뮬레이터의 처리도

본 논문에서는 위의 두 가지 방식이 지원되며 신호대 간섭비 분석과 용량 계산을 할 때는 Snap-Shot 시뮬레이션으로 하고 호 수락제어 알고리즘을 검증할 때는 Real-Time 시뮬레이션을 사용한다.

4. 시뮬레이터의 동작

계층셀 시뮬레이터는 비주얼 C++ 6.0 MFC(Microsoft Class Foundation)를 사용해 다중윈도우 및 모달 다이얼로그 방식으로 개발하였다. 이 프로그램은 70여개의 파일의 1.8MBytes 소스 코드로 구성되어 있으며 실행 파일 크기는 84KBytes이고 Pentium-IV PC의 윈도우즈 환경에서 구동된다. 또한 이 시뮬레이터는 여러 개의 인스턴스를 동시에 구동할 수 있어 조건을 달리 함으로써 여러 결과를 동시에 얻어 낼 수 있다.

<그림 7>은 시뮬레이션 시작 전에 계층 셀의 구성과 동작 모드를 설정하는 창으로 매크로, 마이크로, 그리고 피코 셀의 동작을 임의로 결정할 수 있다. 또한 각 계층마다 기지국의 특성을 지정할 수 있으며 단말기의 동작을 결정할 수 있다. 계층 수는 실제로 존재하는 셀의 겹 수를 결정하며 본 시뮬레이터에서는 중심 셀 밖의 한 계층만 구성하더라도 다 계층 구성과 동일한 결과를 얻을 수 있다[2]. 주파수 개수는 셀룰라 시스템 기지국이 제공하는 주파수 밴드 수를 의미하고 섹터 수는 기지국이 지원하는 섹터 개수를 의미한다. 반복 회수는 Snap-Shot 방식 때의 실행 회수, 수행 시간은 Real-Time 방식 때의 총 실행 시간(초)을 의미한다.

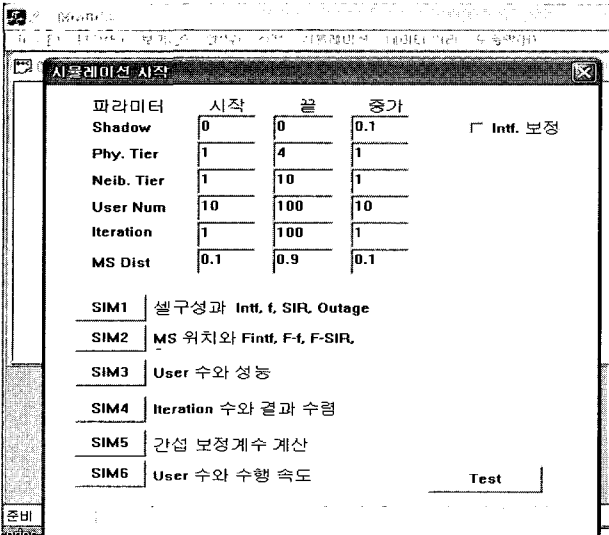


<그림 7> 계층 셀 구성 setup 윈도우

<그림 8>은 여러 가지 성능 지표를 알아보기 위한 시뮬레이션 종류 선택과 파라미터의 초기 값, 최종 값, 그리고 증가량을 설정하기 위한 파라미터 설정 윈도우다. 프로그램 실행 시간과 동작 현황을 파악할 수 있는 상태 바 기능이 있어 조건에 맞지 않을 경우 수행 정지를 할 수 있다.

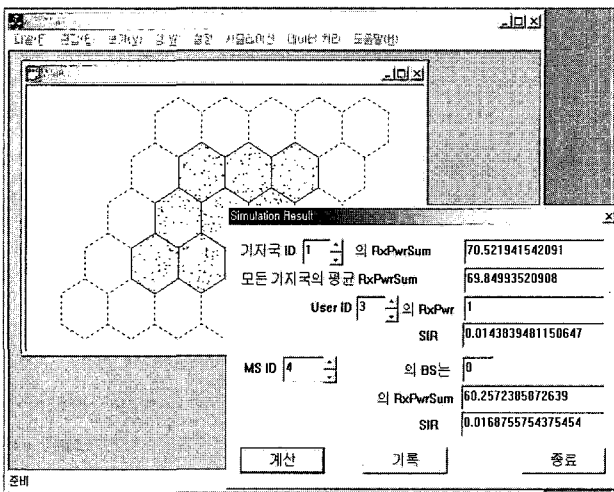
Shadow는 웨도우잉 감쇄 정도, Phy. Tier는 실존하는 기지국의 외곽 겹 수, Neib. Tier는 간섭 계산에 사용할

외각 셀 겹 수, User Num은 단말기 수, Iteration은 반복 수행 회수를 의미하며 각각 시작 값과 끝 값 그리고 증가 값을 설정하여 변화에 따른 성능 결과를 반복 실험하기 위한 수치이다. SIM1~SIM6은 원하는 성능 지표를 얻기 위한 각각의 시뮬레이션 프로그램을 시작하는 버튼이다. 현재는 관심 있는 파라미터 변화 설정과 프로그램을 수동으로 실행하도록 되어 있다.



<그림 8> 시뮬레이션 파라미터 설정 윈도우

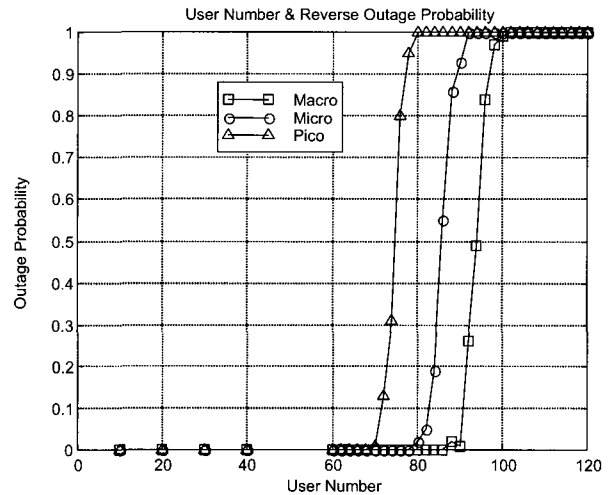
Intf. 보정 체크박스는 Neib. Tier 수가 적더라도 충분한 이웃 셀을 고려한 것과 동일 한 결과가 나오도록 간섭량을 수치적으로 보정해주는 기능의 사용 여부를 결정한다. 이 기능을 사용할 경우 주변 셀 겹 2 이상이면 동일한 결과가 나오므로 시뮬레이션 시간을 대폭 줄일 수 있다[2].



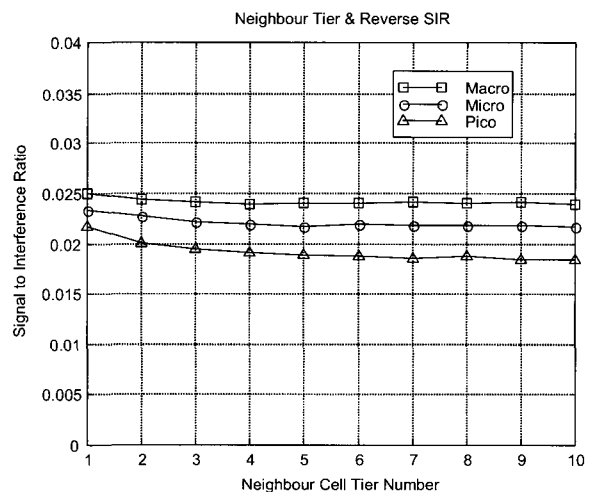
<그림 9> 시뮬레이션 결과 확인 윈도우

<그림 9>는 시뮬레이션이 완료된 후 결과를 확인해 볼 수 있는 윈도우로 모든 기지국과 단말기에서의 측정 결과를 알아볼 수 있다. 확인 대상 기지국의 ID(기지국 일련번호)를 선택하면 RXPwrSum 창에 수신 전력의 합이 출력된다. User ID(사용자 일련번호)를 선택하면 사용자의 수신 전력 크기와 역방향 SIR(신호대 간섭비) 값이 계산되어 출력된다. MS ID(단말기 일련번호)를 선택하면 순방향 수신 전력의 크기와 SIR 값이 출력된다.

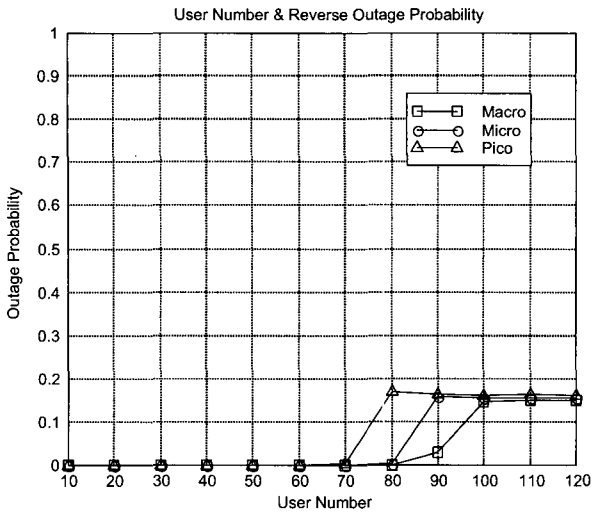
<그림 10>은 개발된 시뮬레이터를 이용하여 계층 셀 시스템을 시뮬레이션 한 특정 결과를 보여준다. 이는 시뮬레이터에서 생성된 데이터 파일을 별도의 그래프 생성도구(MATLAB)를 사용하여 출력한 것이다. 매크로, 마이크로, 피코 셀에 사용자수가 약 90명, 80명, 70명 일 때 Outage 확률이 급격한 증가를 보여 한계 용량을 알 수 있다.



<그림 10> 역방향 Outage 확률과 시스템 용량



<그림 11> 이웃 셀 겹 수에 대한 계층 셀의 역방향 SIR



<그림 12> 호수락 제어 시의 outage 확률

<그림 11>은 셀 구성 설정 시 간섭에 고려할 이웃 셀의 겹 수를 변화시키면서 역방향 SIR을 측정함으로써 매크로 셀은 3겹, 마이크로 셀은 4겹, 피코 셀은 6겹 이상 설정해야 정확한 시뮬레이션이 됨을 알 수 있다. 이는 interference 보정 기능을 사용하지 않은 경우이다. 보정 계수는 이웃 셀을 10겹 이상 최대 구성하여 시뮬레이션 한 후 얻은 간섭 값을 측정하여 나머지 셀 간섭율을 계산한 후 얻을 수 있다[2].

<그림 12>는 본 시뮬레이터를 이용해 CAC 알고리즘을 적용했을 때 얻어진 결과이다. 매크로 셀, 마이크로 셀, 피코 셀 각각 사용자 수가 70명, 80명, 90명 이상 일 때도 outage 확률이 0.2 이상 증가하지 않아 CAC 알고리즘 성능이 나음을 알 수 있다[1].

5. 결론

본 논문에서는 미래의 이동통신망 구성 시 필수적인 계층 셀 구조의 셀룰라 시스템에서의 동작과 성능을 평가하기 위한 시뮬레이터를 설계하고 개발하였다. 계층 셀은 매크로, 마이크로 피코 셀이 중첩되는 일반화된 구성을 적용했으며 시뮬레이션 시간을 최소화하기 위해 wrap-around 기법을 사용했다. 프로그램은 C++ 언어를 사용한 객체지향프로그래밍 기법으로 구현했고 PC에서 구동된다.

개발한 계층 셀 시뮬레이터는 기존의 방식에서 셀 구성 고정된 점을 개선하여 셀 구성을 원하는 데로 변화시킬 수 있으며 wrap-around 처리 범위를 임의로 설정할 수 있어 다양한 셀 구성에서의 시스템 성능을 분석이 가능하다. 또한 실존하는 셀의 겹 수를 하나로 하고 간

섭에 고려할 이웃 셀 겹 수를 2로만 하여도 10겹 이상 고려한 것과 동일한 결과를 얻을 수 있어 시뮬레이션 시간을 1/100로 줄일 수 있다. 그리고 객체지향 기법으로 개발된 클래스는 이식성이 우수해 기능의 확장 및 용도 전환 개발이 용이하다.

개발된 프로그램은 다양한 계층 셀 구성에 따른 시스템의 간섭량, 신호대 간섭비, 용량 등을 평가할 수 있으며 전력 제어 및 호 수락 제어 기법을 적용했을 때의 성능을 평가할 수 있는 도구로 활용될 수 있다.

향후에는 별도의 그래픽 처리가 필요 없이 바로 결과 그래프를 얻을 수 있고, 정해진 시나리오대로 파라미터 변화에 따라 모든 시뮬레이션을 자동으로 실행할 수 있도록 기능을 확장할 계획이다. 또한 이 시뮬레이터를 활용하여 계층 셀에서의 호 수락 제어, 핸드오버 알고리즘, 무선 자원 관리 등의 연구를 수행할 예정이다.

참고문헌

- [1] 김호준; "CDMA 계층 셀 시스템에서의 호 수락 제어에 관한 연구", 한국해양정보통신학회논문지 게재 예정, 10(9), 2006.
- [2] 김호준; "Wrap Around 기법에 의한 CDMA 시스템 시뮬레이션 효율에 관한 연구", 한국통신학회논문지, 28(5) : 276-283, 2003.
- [3] A. Ganz, C. M. Krishna, D. Tang, Z. J. Haas; "On optimal design of multitier wireless cellular systems," *IEEE Communications Magazine*, 35(Issue 2) : 88-93, Feb. 1997.
- [4] Antonio Irea, Antonella Molinaro, and Salvatore Marano; "Wireless Broadband Applications : The Tele-service Model and Adaptive OoS Provisioning," *IEEE Communications Magazine*, pp. 71-75, Oct. 1999.
- [5] Avinash Jain; "Wrap-around for RL-DV Evaluation System Simulation," 3GPP2-TSG-C-20020708-030, July 2002.
- [6] Jens Zander, S. L. Kim; *Radio Resource Management for Wireless Networks*, Artech House Publishers, pp. 336-337, 2001.
- [7] Lauro Ortigoza-Guerrero and A. Hamid Aghvami; *Resource Allocation in Hierarchical Cellular System*, Artech House, pp. 1-6, 2000.
- [8] Liu, X. and Fapojuwo, A. O.; "Resource management for handoff traffic in hierarchical cellular networks," 24th IEEE International Performance, Computing, and Communications Conference, pp. 615-620, April 2005.
- [9] Sallabi, F., Lakas, A., Shuaib, K., Boulmalf, M.;

- “WCDMA downlink simulator with efficient wrap-around technique,” *Wireless and Optical Communications Networks*, Second IFIP International Conference Proceedings, pp. 472-476, March, 2005.
- [10] V. J. Garg and J. E. Wilkes; *Wireless and Personal Communications Systems*, Prentice Hall PTR, pp. 1-47, 1996.
- [11] X. Lagrange; “Multitier Cell Design,” *IEEE Communications Magazine*, 35(Iss. 8), pp. 60-64, Aug. 1997.
- [12] Xiaoxin Wu and Murherjee, B., and Ghosal, D.; “Hierarchical architectures in the third-generation cellular network,” *IEEE Wireless Communications*, 11(Iss. 3) : 62-71, June 2004.
- [13] Young Yoon and Long Duan; “Basic operation of the wrap-around technique for system-level simulation,” 3GPP2-C30-20020708-030, July 2002.