

# 객체지향적 시뮬레이션플랫폼을 이용한 이동통신 시뮬레이터 구현

윤 영 현\* · 김 상 복\*\* · 이 정 배\*\*\*

## 요 약

오래 전부터 이동통신 서비스를 위한 통신 프로토콜이나 기능 요소의 성능을 평가하고 시험하기 위하여 시뮬레이션을 통한 평가 방법이 사용되었다. 본 논문에서는 객체지향적 시뮬레이션 플랫폼을 이용하여 이동통신 서비스의 성능을 예측할 수 있는 이동통신 시뮬레이터인 PCSsim(Personal Communication System Simulator)을 구현하였다. PCSsim은 사용자 이동성, 호 발생 주기 및 호 지속 시간을 고려하여 기지국과 이동호스트를 시뮬레이션할 수 있다. 이를 이용해 본 논문에서는 주거 및 상업지역에서의 통화 발생률, 사용자 이동 속도와 통화 시간에 따른 핸드오프 발생율을 시뮬레이션 결과로 제시하였으며, 시뮬레이션 결과로 발생된 핸드오프율과 실제 서비스 환경에서의 핸드오프율이 거의 유사한 특성을 나타내는 것을 확인할 수 있었다. PCSsim은 설계 과정에서 동적 핸드오프 버퍼링이나 사용자 통화 특성에 따른 기지국 특성을 조정하는데 사용될 수 있으며, 기지국 지역 특성과 사용자 이동성을 반영하여 새로운 이동통신 네트워크를 구축하는데 사용할 수 있다.

## An Implementation of the Mobile Communication Simulator using a Object-Oriented Simulation Platform

Younghyun Yoon\* · Sang Bok Kim\*\* · Jeong Bae Lee\*\*\*

## ABSTRACT

Traditionally, simulation method was used to test and evaluate the performance of communication protocol or functional elements for mobile communication service. In this paper, PCSsim(Personal Communication System Simulator) was realized that can evaluate and review the call process of mobile communication service or to predict its performance by using the object-oriented simulation platform. PCSsim can simulate the base station and mobile host by considering the user's mobility, call generation rate and call duration time. In this paper, based on the simulation, presented the simulation results of hand-off generation ratio according to call generation, user's moving speed and call duration time both in residence area and commercial area, and it was confirmed that the hand-off rates in simulation and actual service environment have similar features. PCSsim can be used in adjusting the characteristics of base station following the dynamic hand-off buffering or the characteristics of user's call in the design stage, and also can be used in building new mobile communication network by reflecting the characteristics of region where the base station is located and the mobility of the user.

**키워드 :** 이동통신(Mobile Communication), 시뮬레이터(Simulator), 객체 지향적(Object Oriented), 트래픽 모델(Tele-Traffic Model)

### 1. 서 론

사용자가 이동하면서 통화하는 이동통신 시스템을 운영하기 위해서는 시스템 구성 요소들의 동작 특성 및 성능을 사전에 충분히 고려하여 시스템을 구축하여야 하므로, 이동통신 시스템의 성능을 평가하고 시험하기 위한 다양한 도구들이 개발되어 사용되고 있다.

이동통신 시스템을 시뮬레이션하기 위해서는 오래 전부터 이동통신 서비스 요청은 포아송(Poisson) 분포를 이용하고 통화 점유 시간은 지수분포를 사용하는 분석적 방법을[1, 2] 사

용하였으나, 분석적 방법에서는 이동호스트의 이동성과 복잡한 이동통신 시스템 내부의 상호 관계를 반영하여 분석하기가 매우 어렵다. 이와 같은 복잡한 환경을 분석하고 평가하기 위해서는 분석적 방법보다는 실제 이동통신 시스템과 유사한 가상 시스템을 만들어 실제 동작 환경에서의 성능을 분석하고 평가하는 시뮬레이션 방법이 요구된다.

이동통신 시스템을 위한 시뮬레이터로는 핀란드의 Helsinki University of Technology에서 GSM 네트워크를 시뮬레이션하여 시각적으로 네트워크의 동작 상태를 보여 주는 MobSim(Mobile Phone Simulator)[3]을 개발하였다. MobSim은 이동호스트의 위치 갱신, 이동호스트 또는 네트워크에 의한 호 시작/ 호 종료 등의 기능이 구현되었다. 또한, GSM 네트워크에서 전송되는 프로토콜의 동작을 한눈에 볼 수 있도록 하

\* 정 회 원 : 명지전문대학 정보통신과 교수

\*\* 정 회 원 : 애니빌 대표이사

\*\*\* 종신회원 : 선문대학교 컴퓨터정보학부 교수

논문접수 : 2004년 1월 9일, 심사완료 : 2004년 7월 28일

기 위하여 시그널링 트래픽을 X-windows를 이용하여 시각적으로 보여 주고 있다. 그러나, MobSim은 시스템의 복잡성을 줄이기 위하여 하나의 이동통신 교환기를 가지고 있는 최소한의 GSM 시스템 환경만을 고려하였고, 이동통신 서비스를 위한 중요한 과정인 핸드오프는 구현되지 않았다. 따라서, MobSim은 GSM네트워크의 동작 방법을 이해하기 위한 교육 지원용 도구로서 사용되고 있다.

또 다른 형태의 시뮬레이터는 독일 Aachen University of Technology에서 개발한 SIMCO3++(Simulation of Mobile COmmunication)[4]이다. SIMCO3++는 GSM, UMTS, DECT와 같은 이동통신 서비스 환경을 위한 mobility, channel modeling, propagation, medium access scheme 등의 기능을 시뮬레이션하여 통신을 위한 각 계층별 프로토콜의 개발 및 성능 측정을 위한 도구로 사용될 수 있다는 장점이 있으며, C++를 사용한 객체지향적 방법에 의하여 개발되었다. 그러나, SIMCO3++는 우리나라에서는 사용하지 않는 통신 방식을 지원하고 있어, 이를 그대로 우리 현실에 적용할 수는 없다는 단점이 있다.

상업용 유무선 통신 시뮬레이터로 BONEs[5], Network II.5 [6] 등과 같은 시뮬레이터가 사용되고 있다. 상업용 시뮬레이터는 기존에 서비스되고 있는 각종 통신 환경에 대한 사전 정의를 포함하고서, 사용자가 시뮬레이션하고자 하는 환경에 맞도록 입력하도록 되어 있어, 시뮬레이터에 이미 반영된 서비스 환경에 대한 시뮬레이션이 아주 간단하면서 정확한 결과를 얻을 수 있다. 하지만 상업용 시뮬레이터는 시뮬레이터에 사전 정의된 각종 시스템 구성 요소의 내용을 사용자가 임의로 변경하는 것이 매우 어려워 사용자는 시뮬레이터에 사전 정의된 기능들만을 사용해야 하는 단점이 있다. 이러한 단점을 보완하기 위해서는 자신이 원하는 부분만을 정교하게 시뮬레이션해 줄 수 있는 독자적인 시뮬레이터가 필요하며, 이렇게 독자적으로 개발된 시뮬레이터들은 시뮬레이션 결과의 정확성을 입증한 이후에 새로운 시스템 개발이나 기존 시스템의 성능을 시험하기 위해 사용되고 있다.

본 논문에서 구현한 PCSsim(Personal Communication System Simulator)은 객체지향적 시뮬레이션 개발 도구인 C++ SIM[7]을 이용하여, 이동통신 서비스에서 중요한 기능을 담당하는 기지국과 이동호스트를 텔레트래픽 모델에 따라 구현한 것이다. PCSsim에서는 기지국별 이동호스트의 이동속도, 이동방향, 통화발생주기, 그리고 통화시간과 같은 통화 특성이 구현되었으며, 핸드오프와 호 단절(Call Drop)과 같은 기지국과 이동호스트간의 통신 절차를 구현하였다. PCSsim은 이동통신 시스템의 기지국 환경 변화에 따른 통화 성능 예측, 기지국 통신 부하 예측 및 핸드오프 알고리즘 개선 등에 사용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 이동통신 텔레트래픽 모델에 따라 정의된 객체클래스를 C++SIM을 이용하여 구현한 PCSsim에 대하여 설명한다. 3장에서는 가상의 도심 환경을 PCSsim을 이용하여 시뮬레이션하고 이에 대한

결과를 분석한다. 4장에서는 PCSsim의 장·단점을 포함한 결론과 향후 연구 방향에 대해서 언급한다.

## 2. 이동통신 시뮬레이터 구현

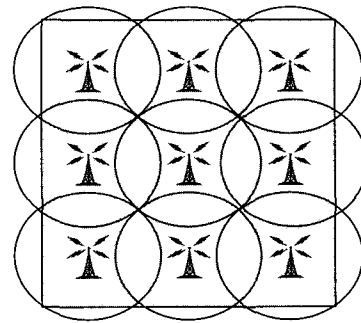
### 2.1 PCSsim 동작 특성

본 논문에서 구현한 PCSsim은 다음과 같은 동작 특성을 갖도록 구현되었다.

첫째, PCSsim에서 기지국의 셀 반경은 같은 크기이며, 기지국은 각 셀의 중심에 위치한다고 가정한다.

둘째, PCSsim의 이동호스트는 네트워크 트래픽 모델에 따라 새로이 생성되며 생성된 이동호스트는 해당 기지국의 트래픽 소스 모델에 따라 이동한다.

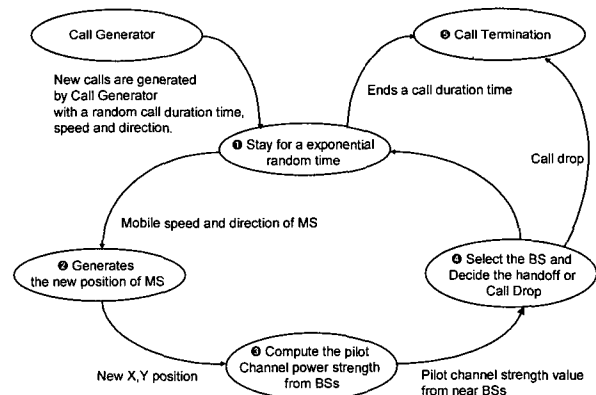
셋째, 시뮬레이터의 기지국은 통화 중 전파 특성, 핸드오프율 및 호 단절율을 시뮬레이션할 수 있도록 실제 운영되는 기지국 환경 변수와 동일한 값을 유지한다.



(그림 1) 시뮬레이션을 위한 이동통신 가상 평면

### 2.2 호 상태 천이(Call State Transition Flow)

(그림 2)는 이동호스트의 각종 상태 변화를 나타내는 호 상태 천이도(Call State Transition Diagram)이며[9], 여기에는 이동호스트의 초기화 상태, 대기 상태, 통화 중 상태, 그리고 시스템 접속 상태를 포함하고 있다. PCSsim에서는 효과적인 시뮬레이션을 위하여 이동호스트가 통화 중인 상태만을 고려하였다[8].



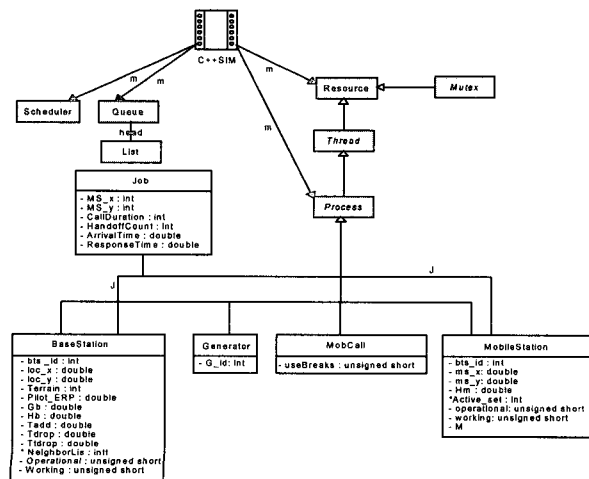
(그림 2) PCSsim 호 상태 천이도

(그림 2)에서 PCSsim의 Call Generator 클래스는 입력으로 주어진 호 발생 주기에 따라 임의의 통화시간, 초기 속도 및 방향을 가진 새로운 이동호스트 객체를 생성한다. 생성된 이동호스트는 ❶상태에서 Exponential 확률분포에 의해 생성된 임의의 시간동안 주어진 속도와 방향으로 이동한다. ❷상태는 이동호스트가 ❶상태에서 이동한 후의 새로운 X, Y 좌표를 구하고, 다시 이동호스트의 이동 속도와 이동방향을 임의로 결정한다. ❸상태는 ❷상태에서 결정된 이동호스트의 새로운 위치와 인접한 기지국으로부터 전송되는 Pilot Channel 전력 세기를 측정한다. ❹상태에서는 ❸상태에서 측정된 강도에 따라 이동호스트가 기존에 통화하던 기지국과 계속 통화하거나, 핸드오프 또는 호 단절 여부를 판단한다. ❺상태는 이동호스트가 다시 ❶상태로 돌아가 임의의 시간동안 위치를 이동하는 과정을 호 단절이나 호 종료시 까지 반복하고, ❻상태로 이동하여 각종 통계 자료를 저장하고 이동호스트 객체를 종료한다.

❶상태에서 이동호스트의 이동성을 시뮬레이션하기 위한 랜덤 지수 분포(Random Exponential Distribution)의 평균 시간이 짧으면 이동호스트의 이동성은 거의 실제 환경과 가깝게 평가될 수 있지만, 동시에 수백 개의 이동호스트 객체가 생성되어 처리되어야 하므로 시스템에 많은 부하가 가해지게 된다. 반대로 시스템 부하를 고려하여 너무 큰 평균 대기 시간을 지정하면 모든 이동호스트는 정지상태에서 통화 중인 것으로 시뮬레이션된다. 따라서, 시뮬레이션 시스템의 성능과 생성된 이동호스트에 따라 평균 대기 시간을 지정하여야 한다. 본 논문에서는 이동 속도가 20km 이상인 이동호스트는 평균 대기 시간을 30초로 지정하고, 20km 이하인 이동호스트는 평균 대기 시간을 60초로 지정하여 시뮬레이션 하였다.

### 2.3 Class Diagram

(그림 3)은 객체지향 설계 도구인 Paradigm Plus[10]를 이용하여 PCSsim을 구성하는 클래스간의 연관 관계를 나타낸



(그림 3) PCSsim 클래스 다이어그램

클래스 다이어그램이다. C++SIM에서 제공하는 시뮬레이션 객체는 회색 바탕으로 표시되어 있다.

(그림 3)에서 Job 클래스는 C++SIM의 시뮬레이션 엔진에서 시간 순서에 따라 수행되어야 하는 작업들을 큐(Queue) 상태로 보관하였다가 처리하는 역할을 수행한다. PCSsim 클래스는 시뮬레이터를 시작하기 위한 메인 클래스이며, Base-Station, Generator 및 MobileStation 클래스가 실제 이동통신 서비스 상태를 시뮬레이션하기 위해 구현된 객체클래스이다.

### 2.4 Call Generator 구현

PCSsim의 Call Generator는 실제 환경의 기지국마다 가해지는 통화 부하가 서로 다른 점을 구현하기 위하여 PCSsim의 기지국마다 Call Generator 객체를 생성시킨다. 생성된 Call Generator 객체는 지정된 기지국의 특성에 따라 포아송 확률분포 함수를 이용하여 시간당 평균 통화 발생 횟수만큼의 이동호스트를 생성한다[8]. 이 때 Call Generator에 의해 생성되는 이동호스트는 기지국별 평균 이동 속도에 따라 초기 속도와 진행 방향이 결정된다. 또한, 지수확률분포 함수에 따라 임의의 통화 시간을 지정받고, 지정된 통화 시간동안 호 상태 천이도에 따라 이동하다가 이동호스트를 종료하게 된다. 이를 통하여 PCSsim에서는 수 많은 이동통신 가입자를 정의하지 않고서, 기지국에서 일정시간 동안 발생하는 실제 통화량만큼의 이동통신 서비스를 시뮬레이션할 수 있게 된다.

<표 1>은 Call Generator 클래스에서 이동호스트 생성과 통화 시간을 결정하기 위해 사용할 확률분포 함수를 결정하는 과정을 구현한 것이다.

<표 1> 이동호스트 생성

```

Generator::Generator (int id, double AvgArrivalRate, double
    AvgCall)
    : G_id(id)
{
    double InterA = 3600/AvgArrivalRate; //호 발생 주기
    double range = AverageCellSize/2; //평균 통화 지속 시간

    //호 발생주기에 따른 랜덤 호 생성
    InterArrivalTime = new ExponentialStream(InterA);
    //평균 통화지속시간에 따른 랜덤 통화지속시간 생성
    Avg_call_duration = new ExponentialStream(AvgCall);
}
    
```

<표 2> 생성된 이동호스트 초기값 지정

```

void Generator::Body ()
{
    for(;;)
    {
        double init_x, init_y; // 새로 생성되는 이동호스트의 기지국내
                               // 최초 x, y, 위치 값
        double call_duration; // 새로 생성되는 이동호스트의 통화 시간
        int Trans_mth; // 이동호스트 이동 수단
    }
}
    
```

```

Hold((*InterArrivalTime)()); // 기지국 위치 및 시간대별
                             // 통화량 발생기
init_x = L_gen->get_Location(); // 이동호스트의 최초 x
                             // 좌표 지정
init_y = L_gen->get_Location(); // 이동호스트의 최초 y
                             // 좌표 지정
call_duration = (*Avg_call_duration)(); // 이동호스트의 통화 시간
                             // 지정
Trans_mth = (*MS_Trans)(); // 이동호스트 이동수단 결정

// 새로 생성된 이동호스트 통화를 시뮬레이터의 Job Queue에 저장
Job* work = new Job(G_id, init_x, init_y, call_duration, Trans_mth);
}
    
```

<표 2>는 <표 1>에서 지정된 확률분포 함수에 따라 이동호스트를 생성하고, 생성된 이동호스트에게 통화 지속 시간과 최초의 위치 및 이동수단을 지정하는 것을 구현한 것이다.

2.5 기지국 구현

PCSSim에서는 기지국과 이동호스트간의 통화 설정 여부를 결정하기 위하여 [8]에서 사용된 Hata모형을 이용하여 이동호스트에서 수신되는 각 기지국의 Pilot Channel의 세기를 결정하고, 이에 따라 이동호스트와의 호 연결, 핸드오프, 그리고 호 단절 여부를 시뮬레이션하도록 구현되었다.

PCSSim에서 기지국을 나타내는 BaseStation 클래스는 기지국이 속한 지역적 특성에 따라 전파 특성이 달라지는 것을 고려하였으며, 이는 사용자 입력에 의해 결정된다. <표 3>은 기지국별 전파 전달 특성을 구현한 것으로 Hata 모델 [11, 12]에서 제시한 함수에 따라 랜덤값을 생성하여 이동호스트에게 제공하도록 구현되었다.

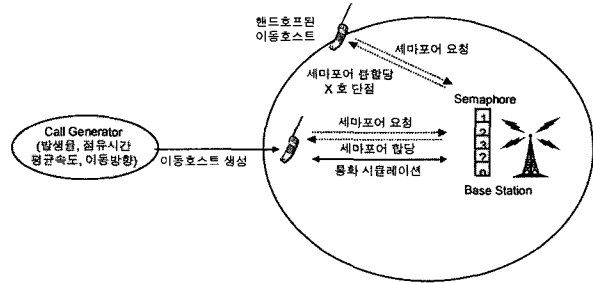
<표 3> 기지국 지역별 전파전달 지연 랜덤값 생성

```

switch(Terrain) {
case 1: // Open Terrain-개발지 전파 전달 특성
    LNI = new NormalStream(0, 0.1);
    R1 = new NormalStream(0, 0.1);
    break;
case 2: // Suburban Terrain-도시주변 지역 전파
        // 전달 특성
    LNI = new NormalStream(0, 0.9);
    R1 = new NormalStream(0, 2.12);
    break;
case 3: // Urban Terrain-도시 지역 전파 전달 특성
    default:
    LNI = new NormalStream(0, 1.82);
    R1 = new NormalStream(0, 4.24);
    break;
}
    
```

이동통신 서비스 중에 발생하는 호 단절은 통신 환경에 의한 영향도 있지만, 기지국에서 동시에 서비스할 수 있는 최대 사용자 수를 초과한 경우에도 호 단절 현상이 일어난다. 이러한 환경을 시뮬레이션하기 위하여 PCSSim에서는 새로이 생성되었거나 핸드오프된 이동호스트 객체가

기지국 객체로부터 서비스를 받기 위해서는 세마포어를 할당받아야만 통화가 연결되도록 하여, 기지국의 통화 채널 부족 현상을 시뮬레이션할 수 있도록 구현하였다. (그림 4)는 PCSSim에서 Call Generator, 이동호스트 및 기지국간의 관계와 세마포어의 역할을 나타낸 것이다.



(그림 4) Call Generator, 이동호스트, 기지국간의 관계도

<표 4>는 기지국이 이동호스트로부터 서비스 요청을 받았을 때, 서비스할 수 있는 여분의 채널이 있는지 여부를 확인하기 위한 세마포어를 구현한 것이다.

<표 4> 기지국에서 이동호스트에 채널 할당 구현

```

if(!sema_trywait(&Semx[bts_id])) { //세마포어 할당 요구
    MobileStation* MS1 = new MobileStation(bts_id, J);
    //세마포어 할당

#ifndef NO_RESOURCE
    Resource::ref(MS1); //시뮬레이션 자원 할당
#endif
    MS1->Activate(); //객체 실행

    J = 0;
} else { //세마포어 할당 실패
    cout << bts_id << "Channel Full !!!\n";
    //사용자에게 통보
    TotalBlockedCalls++; //통계 처리
    BlockedCalls[bts_id]++;
    J = 0;
}
    
```

2.6 이동호스트 구현

이동통신 시스템을 시뮬레이션하기 위한 중요한 요소는 이동호스트이다. PCSSim은 [8]에서 정의한 텔레트래픽 모델에 따라 사용자 이동성을 결정하도록 구현되었다. 식 (1)과 식 (2)는 이동호스트의 속도를 결정하기 위한 함수이다. 식 (1)에서  $S_0$ 는 이동호스트의 최초 속도이며, 식 (2)의  $S_i$ 는 시간변화에 따라 변경된 이동속도이다.

$$S_0 = G(e, \sigma) \tag{1}$$

$$S_i = U[S_{i-1} \times 0.9, S_{i-1} \times 1.1] \text{ where } i = 1, 2, 3 \dots \tag{2}$$

식 (3)과 식 (4), 식 (5)는 이동방향을 결정하기 위한 함수로써, 식 (3)의  $D_0$ 는 이동호스트의 최초 방향이며, 식 (4), 식 (5)의  $D_i$ 는 시간변화에 따라 변경된 이동방향이다.

$$D_0 = U[\phi_1, \phi_2] \quad (3)$$

$$D_i = U[D_{i-1} - \delta, D_{i-1} + \delta] \text{ where } i = 1, 2, 3 \dots$$

for Mobile Host in  $\geq 20\text{km}$  (4)

$$D_i = U[0, 359] \text{ where } i = 1, 2, 3 \dots$$

for Mobile Host in  $< 20\text{km}$  (5)

<표 5>는 PCSsim에서 이동호스트의 속도 및 방향을 결정해 주기 위한 함수를 구현한 것이다.

<표 5> 이동호스트의 속도 및 방향 결정 함수

```
float Get_MS_speed_n_direction()
{
    // 이동호스트의 새로운 방향 결정
    if (MS_Speed < 20) {
        MS_Direction = new UniformStream(0, 359);
    } else {
        MS_Direction = new UniformStream(MS_Direction-δ,
                                         MS_Direction+δ);
    }

    // 이동호스트의 새로운 속도 결정
    MS_Speed = new UniformStream(MS_Speed × 0.9,
                                 MS_Speed × 1.1);
}
```

<표 6>은 PCSsim에서 새로이 생성된 이동호스트를 위한 각종 변수를 초기화하는 것을 구현한 것이다.

<표 6> 이동호스트 변수 초기화

```
// 이동호스트 관련 변수 초기화
CallDrop = FALSE;
HandOFF = FALSE;

working = TRUE;
OngoingCalls++;
OngoingCallsInBTS[bts_id]++;

// 이동호스트의 잔여 통화 시간 변수 초기화
RemainTime = J → get_duration();

// 이동호스트의 진행 방향 변수 초기화
MS_direction = (*DIR_var)();

// 이동호스트의 진행 속도 변수 초기화
MS_Speed = new GaussianStream(L_speed_limit, H_speed_limit);

//(그림 4)의 3가지 기지국 Zone 결정 변수 초기화
Tadd = BTS[bts_id] → get_Tadd();
Tdrop = BTS[bts_id] → get_Tdrop();
Ttdrop = BTS[bts_id] → get_Ttdrop();
```

<표 7>은 PCSsim에서 통화 중 사용자가 이동하는 것을 시뮬레이션하기 위해 구현한 것으로 (그림 2)의 ㉓번과 ㉔번 과정을 구현한 것이다.

<표 8>은 PCSsim에서 이동호스트와 기지국간의 전파 특성을 시뮬레이션하기 위해 기지국으로부터 전달된 Pilot Channel의 강도를 계산하여 핸드오프 및 호 단절 여부를 결

정하도록 (그림 2)의 ㉓번과 ㉔번 과정을 구현한 것이다.

<표 7> 이동호스트의 통화 중 이동 과정 구현

```
do {
    // (그림 2) 이동호스트 상태천이도의 ㉓ 상태
    stay = StayTime();

    if (RemainTime < stay)
        stay = RemainTime;

    // Job Queue에서 stay 시뮬레이션 시간이 경과하는 동안 대기
    Hold(stay);

    RemainTime -= stay;

    // (그림 2) 이동호스트 상태천이도의 ㉔ 상태
    // 통화 잔여 시간이 존재하는 경우, 이동후의 새로운 위치 좌표 계산
    if (RemainTime > 0) {
        Compute_position(stay, speed, direction, var_x, var_y);

        //이전 좌표에서 변경된 좌표를 반영하여, 새로운 좌표 계산
        ms_x = ms_x + var_x;
        if(ms_x < 0)
            ms_x = 0;
        ms_y = ms_y + var_y;
        if(ms_y < 0)
            ms_y = 0;

        //통화 잔여 시간이 존재하는 경우, 이동후의 새로운 속도, 방향 계산
        Get_MS_speed_n_direction();
    }
} while(RemainTime > 0);
```

<표 8> 이동호스트의 Pilot Channel 세기 측정 및 핸드오프 결정

```
// (그림 2)의 ㉓ 상태
// 현재 위치에서 기지국과의 Pilot Channel Strength 검사
new_bts = Compute_PCSM(this, Tadd);
if((Active_set[0] == -1) && (Active_set[1] == -1) &&
    (Active_set[2] == -1)){
    // Active_set 안에 기지국이 없어 호 단절
    CallDrop = TRUE;
    break;
}

// (그림 2)의 ㉔ 상태
// 핸드오프 발생 여부 검사
if((bts_id != new_bts) && (Active_set[1] != bts_id) &&
    (Active_set[2] != bts_id)){
    TotalHandoffs++;
    Handoffs[new_bts][bts_id]++;
    IncomingHandoffs[new_bts]++;
    OutgoingHandoffs[bts_id]++;

    // 핸드오프된 기지국에 세마포어로 주파수 할당 요청
    sema_post(&Semx[bts_id]);
    bts_id = new_bts;
    if(sema_trywait(&Semx[bts_id])) {
        // 할당할 주파수가 없어 호 단절
        CallDrop = TRUE;
    }
    // 이동호스트가 보유한 세마포어 반환
    sema_post(&Semx[bts_id]);
    // 이동호스트 객체 삭제
    this → terminate();
    break;
}
}
```

### 3. PCSSim 실험 결과 및 분석

#### 3.1 시뮬레이션 환경

구현된 PCSSim의 성능을 시험하기 위하여 기지국이 설치되어 있는 지역이 상업지역과 주거지역이라고 가정하고, 서울시 지역별 평균 속도[13]와 통화 특성을 <표 9>와 같이 정하고, (그림 5)과 같은 텍스트 파일 형태로 PCSSim을 실행하기 위해 요구되는 변수 값을 입력하였다.

<표 9> 기지국 입력 변수 값

변수명	설 명	설 정 값
S <sub>b</sub>	셀내 지하철 통과 여부	On/Off
기지국 크기	시뮬레이션을 위한 가상 공간 크기	5×5Km정방형 구조
V <sub>avg</sub> (km/h)	셀 내의 차량 이동속도	17.72(도심승용차)/20.94(도심버스) 25.90(외곽승용차)/19.13(외곽버스)
InterA	호 발생주기(calls/hour)	3,042(상업) 1,850(주거)
Dtime	호 점유시간(sec)	65

(그림 5)은 PCSSim을 위한 입력 파일인 "PCSSim.dat"의 예제이다. 입력 파일은 모든 기지국 및 시뮬레이션을 위한 공통 변수 부분과 각 기지국별 특성 변수 부분으로 구성되어 있다.

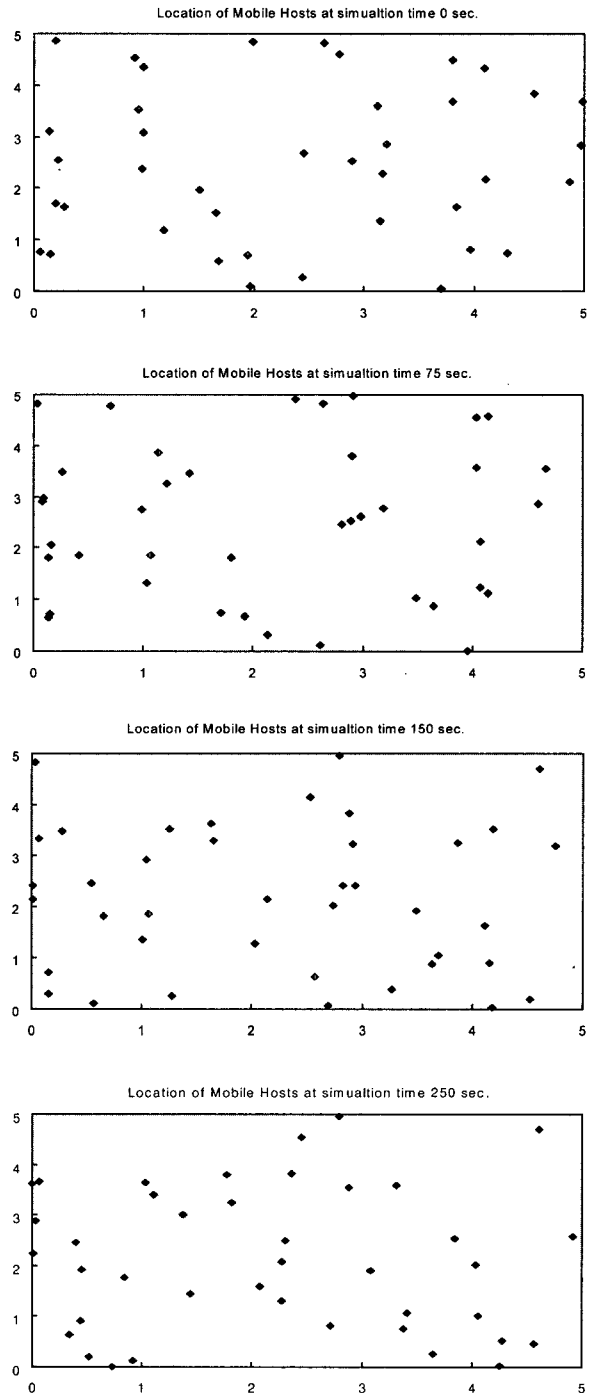
기지국 공통 특성 변수	SimulationTimes = 6hours LpMethod = 1 1 : Hata_model AverageCellSize = 5Km TotalTrafficChannelsERP = 57dBm MobileERP = 20dBm NumberOfUsers = 200users TrafficChannelActivityFactor = 0.6unit PilotChannelERP = 51.5dBm PagingChannelERP = 46.94dBm SyncChannelERP = 41.5dBm BaseStationNoiseFigure = 5dB MobileNoiseFigure = 8dB ForwardTrafficChannelBitRate = 9600bps ReverseTrafficChannelBitRate = 9600bps PagingChannelBitRate = 9600bps SyncChannelBitRate = 9600bpsBandwidth = 900Mhz ChipRate = 1.2288Mchips MobileAntennaGain = 0dB AverageMobileHeight = 1.5m FrequencyReuseFactor = 0.65unit
기지국별 특성 변수	NumberOfBaseStation= 36 unit 0 5 5 2 51.4 14 30 -14.0 -18.0 5.0 1 6 7 -1 -1 -1 -1 0 35 0 359 397.8 100 1 5 10 2 51.4 14 30 -14.0 -18.0 5.0 0 2 6 7 8 -1 -1 -1 035 0 359 397.8 100 ..... ..... ..... 35 5 10 2 51.4 14 30 -14.0 -18.0 5.0 0 2 6 7 8 -1 -1 -1 0 35 0 359 397.8 100

(그림 5) PCSSim.dat 파일 예제

#### 3.2 실험 결과 및 분석

##### 3.2.1 이동호스트의 이동

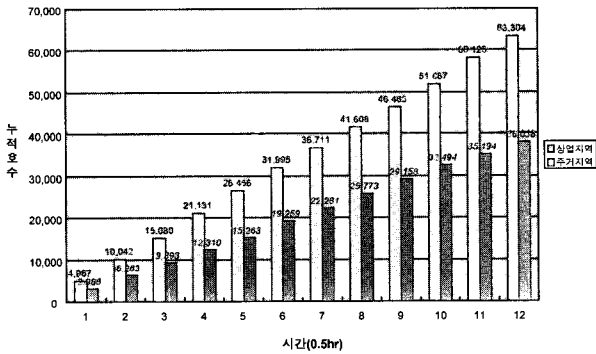
(그림 6)은 앞에서 정의된 이동호스트의 이동 특성에 따라 시간별로 이동호스트의 이동을 나타낸 것이다. (그림 6)에서 각 이동호스트는 주어진 공간 내에서 uniform 분포에 의하여 초기 위치가 결정되며, 시뮬레이션 시간이 경과하면서 각 이동호스트의 이동성(mobility)에 따라서 주어진 평균 속도와 방향으로 이동하는 과정을 나타내고 있다.



(그림 6) 시간대별 이동호스트 위치 이동 결과

### 3.2.2 호 발생

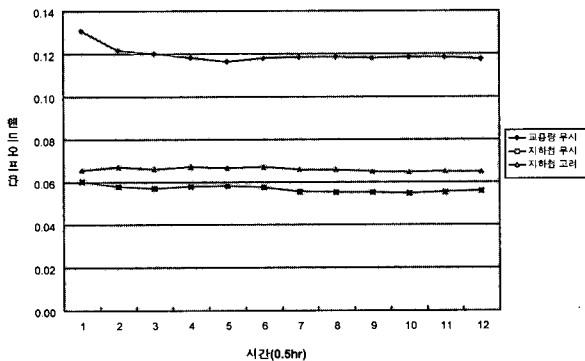
시뮬레이션에서 발생하는 상업지역과 주거지역의 호 누적 개수는 (그림 7)과 같다. (그림 7)에서는 상업지역에서 발생하는 호 개수가 주거지역에 비하여 보다 많이 발생되어 처리됨을 보여주고 있다. 실제 서비스 환경에서도 상업지역에서 요구되는 이동호스트의 통화 및 부가서비스 요구가 주거지역에 비하여 많으므로, PCSsim은 기지국 설치 지역의 호 발생 특성이 반영된 시뮬레이션 결과가 제공된다는 것을 확인할 수 있다.



(그림 7) 지역별 누적 통화 발생 수

### 3.2.3 차량 이동 속도에 따른 핸드오프 분석

(그림 8)은 상업지역에서 교통량을 고려하지 않았을 때와 [8]에서 제시한 트래픽 소스 모델에 따른 교통량과 이동속도를 반영했을 때의 핸드오프율을 분석한 것이다. 교통량을 고려하지 않은 시뮬레이션은 셀 내의 평균 이동 속도만을 반영한 결과이며, 트래픽 소스 모델을 적용한 결과는 [8]에서 제시한 정지 및 저속으로 움직이는 이동호스트와 셀 지역 내에서의 승용차, 버스, 지하철의 이동속도와 수송분담율이 반영된 것이다.

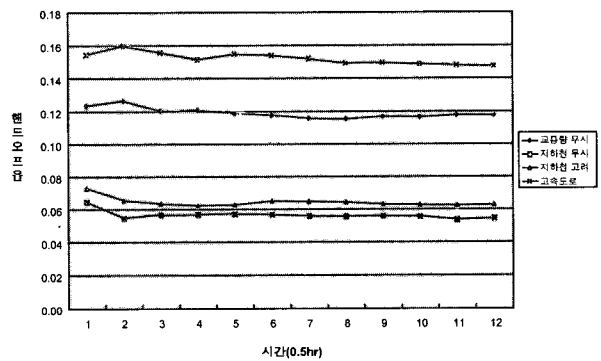


(그림 8) 상업지역에서 교통량에 따른 핸드오프율

(그림 8)에서 교통량을 고려하지 않은 경우의 핸드오프율은 모든 이동호스트가 평균 속도로 움직이는 것으로 시뮬레이션되기 때문에 교통량을 고려한 경우보다 높은 핸드오프율을 보여준다. 또한, 트래픽 소스 모델에서 지하철을 고려

한 경우의 핸드오프율이 지하철의 지향성과 차량 이동속도에 의해 지하철이 고려되지 않았을 때보다 높아짐을 보이고 있다.

(그림 9)은 주거지역과 고속도로가 있는 기지국에서의 핸드오프율을 시뮬레이션한 결과이다. 고속도로는 주거지역과 다르게 고속도로가 설치되어 있는 방향에 따라 양방향 고속으로 차량이 이동하는 특성을 가지고 있다. 따라서, 주로 정지 또는 저속으로 움직이는 특성을 가진 기지국에 비하여 2배 정도 많은 핸드오프율을 보여주고 있다. PCSsim에서는 이러한 기지국의 특성 변화를 간단한 입력에 의하여 시뮬레이션할 수 있다.

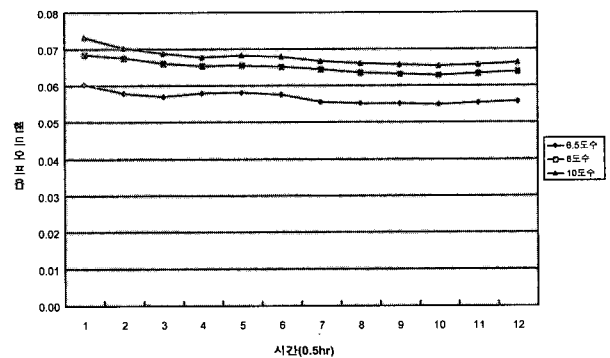


(그림 9) 고속도로와 주거 지역에서의 핸드오프율 비교

### 3.2.4 호 점유 시간에 따른 핸드오프 분석

핸드오프는 이동호스트가 통화 중에 이동하면서 서비스를 제공받는 기지국이 변화하는 것으로 이동호스트의 통화 중 이동거리와 밀접한 관계가 있다. 여기서 이동호스트 이동거리의 평균 이동속도와 이동시간이 되므로 이동통신 호 처리에서 핸드오프율이 변화할 수 있는 또 다른 요소는 이동호스트의 평균 통화 시간이다.

(그림 10)은 상업지역의 평균 통화 시간에 따른 핸드오프율의 변화를 보여주고 있다. 실제 이동통신 환경에서 평균 통화 시간은 6.5도수(1도수 = 10초)이며, (그림 10)은 이를 비교한 것이다.



(그림 10) 평균 호 점유시간에 따른 핸드오프율 비교

(그림 10)에서는 평균 통화 시간이 증가에 따라 보다 높은 핸드오프율이 나타나고 있다. 이는 이동통신 서비스가 기존 음성 위주의 서비스에서 무선 인터넷을 비롯한 다양한 무선 데이터통신 서비스 위주로 변화하면서 평균 통화 시간이 증가하고 있으므로, 기존의 기지국에서 음성 서비스 위주의 핸드오프를 위한 각종 파라미터를 변경 및 보완하는 작업이 요구됨을 알 수 있다.

#### 4. 결 론

본 논문에서는 객체지향적 시뮬레이션 개발 도구인 C++ SIM을 활용하여 이동통신 시뮬레이터인 PCSsim을 구현하였다. PCSsim은 [8]에서 제시한 기지국의 설치 지역별 호 발생 특성이 반영된 네트워크 트래픽 모델과 각 기지국 설치 지역별 차량 평균 속도와 수송분담율을 반영한 트래픽 소스 모델이 적용되었다. 또한, 세마포어를 사용한 기지국 통화 채널 할당 기능을 구현하였고, Pilot Channel 전력 세기 계산 기능을 이용하여 핸드오프율을 결정할 수 있도록 하였다.

본 논문에서 구현된 PCSsim을 이용하여 도심의 주거지역과 상업지역에서의 호 발생량 비교와 다양한 이동 및 통화 환경 변화에 따른 핸드오프율을 결과로 제시함으로써, PCSsim의 시뮬레이션 결과가 실제 환경에서의 호 동작 특성과 유사하다는 것을 보여 주었다. 이를 통해 PCSsim을 기지국에서 이동호스트 호 패턴에 따른 동적 핸드오프 버퍼링과 같은 기능적 요소를 설계 단계에서 고려하거나, 성능 시험에 사용할 수 있으며 이동호스트 이동성과 지역 특성을 고려한 셀 디자인에 이용될 수 있다.

본 논문에서 구현된 PCSsim에서 추가적으로 고려해야 할 요소는 이동통신 서비스에서 늘어나는 데이터 통신을 반영하기 위하여, 데이터 통신 특성이 반영된 텔레트래픽 모델을 정의하고 이를 구현하는 것이다.

#### 참 고 문 헌

[1] Nail Akar, Nihat Cem Oguz and Khsrow Sohraby, "TELP-LACK : An Advanced Teletraffic Analysis Package," IEEE Communication Mag., Aug., 1998.  
 [2] Huro Akimaru and Konosuke Kawashima, "Teletraffic : Theory and Applications," Springer-Verlag, 1993.  
 [3] Hannu Arvelo, "MobSim : A GSM Network Simulator", <http://mordor.cs.hut.fi/tik-76.115/95-96/palautukset/Mobiili/pt/manual.html>  
 [4] Martin Stepler, C. H. Rokitansky, "SIMCO3++ : Simulation of Mobile Communication", [http://www.comnets.rwth-aachen.de/report/subsection2\\_4\\_12\\_4.html](http://www.comnets.rwth-aachen.de/report/subsection2_4_12_4.html)  
 [5] BONes, "Systems & Networks, BONes DESIGNER 3.0 Modeling Guide".

[6] Network II.5, <http://www.caci.com>.  
 [7] M. C. Little, D. L. McCue, "Construction and Use of a Simulation Package in C++," <http://cxsim.ncl.ac.uk/manual/cppsim.html>.  
 [8] 윤영현, "이동통신 시뮬레이터를 위한 텔레트래픽 모델과 객체 클래스 설계," 정보처리학회논문지C, 제11-C권 제4호, August, 2004.  
 [9] Telecommunications Industry Association, "TR-45 Mobile Station - Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System", pp.6-70-72, Ballot Version.  
 [10] Paradigm Plus, <http://www.platinum.com>.  
 [11] Vijay K. Garg and Joseph E. Wilkes, "Wireless and Personal Communications Systems," Prentice Hall PTR, 1996.  
 [12] Vijay K. Garg, Kenneth Smolik and Joseph E. Wilkes, "Applications of CDMA in Wireless/Personal Communications," Prentice Hall PTR, pp.140-143, 1997.  
 [13] 도철웅, "교통공학원론 (상)", pp.59-61.



#### 윤 영 현

e-mail : yhyoon@mail.mjc.ac.kr

1989년 전북대학교 컴퓨터공학과 졸업  
 1994년 고려대학교 전자공학과 공학석사  
 2000년 고려대학교 전자공학과 공학박사  
 1989년~1991년 대우통신 기술부  
 1991년~2001년 SKC&C 기술연구소 과장

2002년~현재 명지전문대학 정보통신과 조교수  
 관심분야 : 멀티미디어 라우팅, Active Network, 무선망, 지능망



#### 김 상 복

E-mail : sbkim@anybil.com

1992년 한양대학교 경영학과(경영학사)  
 2004년 선문대학교 일반대학원 전자계산학과 재학  
 1992년 대우정보시스템 기술기획  
 2000년~현재 애니빌 대표이사

관심분야 : 무선인터넷 저작도구 및 솔루션, 모바일 임베디드



#### 이 정 배

e-mail : jblee@sunmoon.ac.kr

1981년 경북대학교 전자공학과 전자계산 전공(공학사)  
 1983년 경북대학교 대학원 전자공학과 전자계산전공(공학석사)  
 1995년 한양대학교 대학원 전자공학과 정보통신전공(공학박사)

1982년~1991년 한국전자통신연구원 선임연구원  
 1991년~2002년 부산외국어대학교 컴퓨터·전자공학부 부교수  
 2002년~현재 선문대학교 컴퓨터정보학부 부교수  
 관심분야 : 실시간 시스템, 임베디드 시스템, 실시간 통신 프로토콜