

실시간 모니터링 기법을 이용한 모바일 성능 관리 시스템 개발

정택원¹, 이철한², 이종득^{1*}
¹전북대학교 응용시스템공학부, ²(주)엠네스

Development of Mobile Performance Management System Using Real-Time Monitoring Method

Taegwon Jeong¹, Chulhan Lee² and Chongdeuk Lee^{1*}

¹Div. of Applied Sys. Eng., Chonbuk National University, ²mnes

요 약 최근에 모바일 단말기를 이용한 LBS(Location Based Service), MDT(Mobile Data Terminal) 및 PDA(Personal Digital Assistant) 서비스가 일반화 되면서 이들 서비스에 대한 성능을 자동으로 관리하기 위한 시스템이 구현되고 있다. 본 논문에서는 모바일 사용자 필드에서 사용자와 단말기 그리고 단말기와 모바일 서버간의 성능 테스트가 가능하도록 모바일 성능 관리 시스템을 구현하였다. 제안된 시스템은 시나리오 관리 모듈, MDT monitoring, LBS monitoring, PDA monitoring의 4개의 모듈로 구성되며 시나리오 관리 모듈에 의해 각 모듈이 모니터링을 수행하여 성능을 관리하도록 하였다. LBS 모니터링은 LBS 발신 단계, 호 연결 상태, 접속 상태에서의 LBS 테스트 단계를 거쳐 수행하도록 하였으며 그 결과 호의 상태에 따라 성능 관리가 수행되도록 하였다. MDT 모니터링은 입력상태에 따라 command가 수행하도록 구현하였으며, PDA 모니터링은 설정메뉴와 실행메뉴에 따라 모니터링이 수행되도록 구현하였다.

Abstract As the spread of LBS(Location Based Service), MDT(Mobile Data Terminal) and PDA (Personal Digital Assistant) services using mobile terminals, various systems are under development to manage the performance of these services. This paper describes a mobile performance management system which is implemented to test the performance of mobility between the user and the terminal or between the terminal and the mobile server automatically in the mobile-user field. The system has four modules: scenario management module, MDT monitoring, LBS monitoring, and PDA monitoring. The scenario management module manages the performance by monitoring all modules. The LBS monitoring module monitors LBS call attempt stage, call connecting stage, and LBS test for established connection. Thus, the LBS monitoring module manages the performance for every stage of the call. The MDT monitoring module monitors the performance according to the input command and the PDA monitoring module monitors according to the input from the set up and the execution menu.

Key Words : LBS, MDT, Monitoring, Simulator

1. 서론

최근에 무선 및 모바일 통신 서비스의 중요성이 지속적으로 증가됨에 따라 모바일 환경에서 통신 서비스의 성능을 실시간 자동으로 측정 및 관리하기 위한 여러 시스템들이 제안되고 있다. 모바일 성능 관리를 위한 통신

서비스 시스템은 PDA, 모바일 폰, 노트북 등의 디바이스를 기본정보로 하며, 빠른 속도로 그 영역이 확대되고 있다[1-2]. 이러한 인식에 따라 모바일 환경에서 서비스 시스템을 위한 성능 관리 기능 또한 현장 중심으로 빠르게 이동되고 있다. 특히 모바일 통신상에서 멀티미디어 처리, 다중 작업 및 실시간 처리 능력은 시스템 성능 평가

*교신저자 : 이종득(cdlee1008@chonbuk.ac.kr)

접수일 09년 04월 10일

수정일 (1차 09년 06월 30일, 2차 09년 07월 15일)

게재확정일 09년 08월 19일

의 주요 요인이 되고 있으며, 통신 서비스의 실시간 자동 성능 관리 기능은 차세대 모바일 서비스 분야 중 중요한 서비스 기법으로 자리 매김 될 것으로 기대된다[2-4]. 이러한 중요성으로 인하여 정부에서는 경쟁력 있는 이동통신 신성장 국가 산업 달성을 위해서 모바일과 관련한 다양한 신 성장 프로젝트를 계획 중에 있다. 그러나 현재 수많은 사용자를 확보하고 있는 무선통신 및 이동통신 서비스 성능 관리 기능은 단순한 텍스트 중심의 관리가 주를 이루고 있으며, 무선 및 모바일 통신 시대의 어플리케이션을 지원하는 자동 성능 관리 기능과 이기종 기반의 이동통신 환경에서 다양한 콘텐츠를 실시간으로 측정 관리하기 위한 시스템 연구는 미흡한 실정이다. 지금까지 모바일 하드웨어를 위한 SoC, 시스템 소프트웨어를 위한 통신 프로토콜, 프리미티브 GUI, 미들웨어를 위한 스트리밍 관리 및 캐싱 등과 같은 기법들[5-9]은 많은 연구가 진행되고 있으나 시스템 성능 관점에서의 모니터링을 통한 모바일 성능 관리 시스템 개발은 미흡한 실정이다. 최근에는 이러한 문제를 해결하기 위하여 혼합 모니터링 기법[10-12]을 이용한 방법들이 제안되고 있으며, 이 방법은 네트워크의 상태를 파악하여 자원 및 서비스 품질에 대한 성능을 측정하고 관리하는 기법이다. 그리고 [10]에서는 그리드 상에서 패킷의 상태를 측정하여 성능을 관리하는 기법을 제안하였으며 이 방법은 정적 모니터링과 동적 모니터링에 의해 성능이 관리되고 측정된다. [11]에서는 모바일 상에서 응답 패킷의 상태를 파악하고, 이에 따라 종단간(End-to-End)의 경로 상태를 측정 계산하는 기법을 제안하였다. 이들 기법들은 자원 상태를 측정함으로써 네트워크 자원 관리에 대한 효율성은 제공되지만 모바일 단말의 서비스 품질이 실시간으로 관리되지 못하는 문제가 발생한다. 특히 기존의 모바일 성능 관리 시스템에서는 LBS, MDT 및 PDA 성능 관리가 독립적으로 수행되고 있기 때문에 관리의 중복성 문제와 비용문제가 발생되고 있다.

따라서 본 논문에서는 모바일 통신 시스템을 이용하는 서비스 사용자가 언제 어디서나 시공간상의 제약을 받지 않고 이동통신 환경에서 양질의 서비스를 지원받을 수 있도록 모바일 성능 관리 시스템을 구현한다. 구현된 시스템은 LBS(Location Based Service), MDT(Mobile Data Terminal) 및 PDA(Personal Digital Assistant)의 성능 관리를 독립적으로 수행할 수 있고, 모니터링을 통하여 동시에 성능 관리를 수행하도록 하였다. 구현된 시스템은 사용자의 위치를 파악하여 서비스 및 개인 프로파일 성능 관리를 위한 LBS(Location Based Service), LBS 위치 정보 수신을 통해서 전송상태를 모니터링하여 관리하는 MDT(Mobile Data Terminal), 그리고 반복 동작 모니터링

을 통하여 원하는 횟수만큼 이벤트들을 처리하고 관리하는 PDA 성능관리 시스템으로 구성되어 있다. 본 논문의 구성은 다음과 같다. 2장에서는 개발 환경요소에 대해서 살펴보고, 3장에서는 LBS, MDT 및 PDA 모니터링에 기반한 성능 관리 시스템을 설계한다. 그리고 4장에서는 제안된 시스템을 구현하며, 끝으로 결론에 대해 기술한다.

2. 구현 환경요소

2.1 버퍼링

버퍼링(buffer)은 버퍼에 저장된 데이터의 크기를 결정하여 버퍼링을 수행할 데이터를 관리하기 위한 구조이다. 이때 버퍼링이 수행되는 성능관리 버퍼(buffer)는 네트워크 프로토콜이 COM PORT/USB와 TCP/IP 통신이 수행되도록 구성하며, 헤더(header)와 트레일러(trailer), 버퍼 성능 관리 데이터 입출력(B_{in}/B_{out})을 구성하여 성능을 관리한다. 여기서 B_{in} 은 성능 버퍼 관리 구조에서 패킷이 입력되는 입력전송이며, B_{out} 은 패킷이 출력되는 출력전송이다. 버퍼링이 수행되는 메모리 크기는 108바이트 이내로 설정하며, 크기가 108바이트 이상이면 클러스터를 구성하여 버퍼링을 수행해야 한다. 제안된 논문에서 버퍼링이 수행되는 이벤트 패킷 성능 관리는 내부 버퍼 구조와 외부 버퍼구조와의 상호 작용을 통해서 수행되도록 하며, 이벤트가 동시에 처리되지 않도록 구성한다.

2.2 버퍼 루틴

버퍼 루틴은 버퍼에 할당된 이벤트와 성능을 관리하기 위해 사용되며, 할당루틴, 해제 루틴, 이벤트 삭제 루틴, 복사 루틴, 버퍼와 버퍼 간의 이동 루틴 등으로 구성된다. 버퍼루틴을 수행하는 함수는 `m_get()`이며, 버퍼 루틴의 모니터링을 해제하는 함수는 `m_free()`이다. 그리고 버퍼 루틴에서 성능 관리 모니터링을 삭제하는 함수는 `m_delete()`이며, 모니터링을 통해서 관리될 이벤트를 복사하는 함수는 `m_copy()`이다.

2.3 이벤트 처리

버퍼에서 이벤트 처리는 FIFO 큐(Queue)로 구성되며, 큐에는 모니터링을 수행하는 여러 개의 메시지로 구성된다. 버퍼에서 모니터링을 수행하는 메시지는 메시지 블록과 데이터 블록을 통해서 수행되며, 메시지 블록에서는 멀티캐스트가 수행되도록 동일한 메시지를 여러 개의 포트 구성한다. 그리고 메시지의 복사본을 여러 개를 만드는 대신에 메시지를 직접 할당하도록 참조 값을 이용

한다. 참조 값을 이용하여 이벤트 처리를 해제하면 메시지 블록과 연관된 스트림이 해제되는 것이 아니라 이벤트를 수행하기 위해 데이터 블록의 참조 값 필드를 감소시키게 된다. 이벤트 처리는 버퍼가 실행중일 때 동적으로 로드되며, 실행이 중지되면 로드를 해제하여 모니터링을 중단하게 된다.

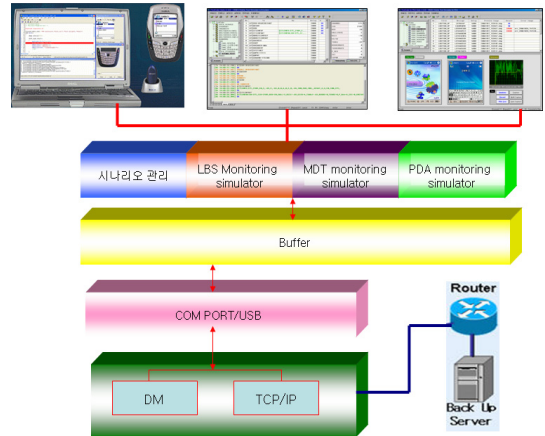
2.4 버퍼 관리

버퍼 관리는 버퍼링, 버퍼루틴, 이벤트처리 구조를 관리하게 되며, 모니터링을 수행하기 위한 이벤트는 하나 이상의 이벤트 블록으로 구성된다. 버퍼 관리 구조에서 이벤트 처리는 버퍼루틴에 의해 제어되며, 이러한 제어 기능은 버퍼와 버퍼 사이의 모니터링을 효율적으로 유지해 주기 위한 기능이다. 즉 모니터링을 수행하는 과정에서 로드가 많이 발생되면 모니터링이 수행되지 않을 수 있으며, 이때에는 해제 기능을 신속하게 수행해야 한다. 그러나 제어 상태를 수행한 후에는 로드 상황을 파악할 수가 있게 되어 모니터링 과정이 정상적인 상태로 돌아오게 된다. 버퍼 관리를 위한 제어는 모니터링에 의해 수행되며, 모니터링이 수행되는 트랜잭션의 수와 오류 상태에 따라 모니터링 활성화가 결정되게 된다.

3. 시스템 설계

이 장에서는 모바일 성능 관리 기능이 기존 수작업으로 하던 방식에서 벗어나 시나리오 별로 Auto_Test 기능 및 성능 관리가 수행하도록 시스템을 설계하며, 제안된 시스템 구성도는 그림1과 같다.

그림1에서 제안된 시스템은 모바일 서비스 목적에 따라 단말기의 성능을 체계화 및 자동화하여 성능 관리 시간을 단축함으로써 특정 버그 발생 시 테스트 반복기능을 수행함으로써 정확한 성능 관리가 수행되도록 한다. 이와 같은 서비스의 자동 성능 관리를 위해 COM PORT 또는 USB 케이블로 연결하여 단말기의 환경에 따라 기지국/중계기 등을 거쳐 호 연결 및 데이터 통신을 수행한다. 데이터 통신과정에서 통신의 접속 상태 및 성능은 버퍼링에 의해 결정되며, 버퍼링의 모니터링을 위해 본 논문에서는 시나리오 관리, LBS monitoring, MDT monitoring, PDA monitoring 기능을 수행한다. 이러한 모니터링 결과는 백업 서버(Backup Server)로 전송되며, 전송 결과에 따라 성능 관리 기능이 결정된다.



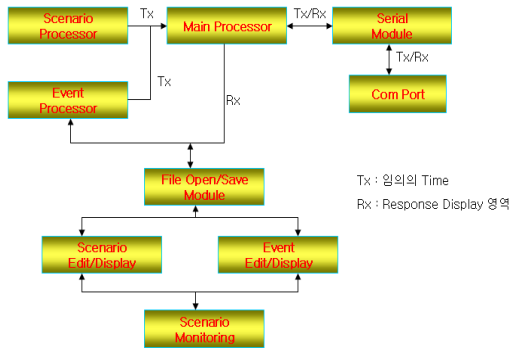
[그림 1] 제안된 시스템 구성도

3.1 시나리오 관리

시나리오 관리기능은 현재 수행중인 모니터링에 대해 시나리오에 포함된 Item을 추가/삭제/수정하는 기능을 제공하며 새로운 이벤트 추가/삭제기능 및 각 이벤트를 선택하여 이벤트에 포함된 Item을 모니터링 하는 기능을 수행한다. 모니터링을 위한 시나리오 관리기능은 편집기능, 자동수행기능, 이벤트 처리기능, 일시정지기능, 송/수신기능 등으로 구성된다. 편집 기능은 해당 Item에 대해 편집창을 이용하여 HEXA 또는 ASCII형태의 데이터를 편집하여 모니터링하기 위한 기능이다. 모니터링 편집은 Tx/Rx로 구분하여 Delay Time과 각 단계에서의 시나리오에 따라 수행된다.

시나리오 자동 수행기능은 처음부터 또는 현재위치부터 시나리오 상태를 자동으로 모니터링 하는 기능이다. 수행단계는 Delay 내에서 수행되면 모니터링은 바로 다음단계로 넘어가 수행되며, 지정된 Delay 내에서 수행되지 않으면 Fail처리를 수행하고 다음 단계로 넘어가게 된다. 이벤트 처리기능은 시나리오 순서에 따라 모니터링을 제어하는 기능이며 예를 들어 전송수신, 메시지 수신 등의 이벤트가 발생하였을 때 모니터링은 순차적으로 실행되며, 실행이 종료되었을 때 이전 시나리오의 이벤트 발생 시점으로 돌아가 모니터링을 다시 실행하는 기능이다. 시나리오 일시 정지 기능은 시나리오에 따라 자동으로 버퍼링을 수행하고 있는 이벤트들에 대해서 모니터링을 원할 정지된 순간부터 다시 모니터링을 수행하는 기능이다. 시나리오 송수신 명령처리 기능은 모니터링을 수행하고 있을 때 송수신되는 명령들의 값이 Log화면을 통해 디스플레이 되도록 하는 기능이며, Log화면에 디스플레이 되는 모든 이벤트들을 추후 Log파일을 이용하여 확인하는 기능이다. 이때 Log 화면은 HEXA 또는 ASCII 값

으로 모니터링되며 시나리오관리를 위한 과정은 그림2와 같다.



[그림 2] 시나리오 관리 프로세스

그림2에서 시나리오 프로세서는 Test하고자 하는 이벤트들을 Tx와 Rx의 절차에 따라 이벤트 명령어와 Ack를 순차적으로 실행하며, 실행이 끝난 후에는 Test 과정을 시행하거나 편집을 하여 다른 이름으로 저장하게된다.

3.2 LBS monitoring

무선인터넷 발달과 응용 대역 확산으로 인하여 위치정보를 이용한 서비스 형태는 다양해지고 있으며 이를 위한 관리 기법 또한 다양해지고 있다[9]. LBS 플랫폼은 GMLC(Gateway Mobile Location Center)[3], MPC(Mobile Positioning Center)[4]등과 같이 위치정보를 제공하는 게이트웨이 서버, 프리젠테이션 또는 라우터, 디렉토리, 유틸리티 서비스 등의 공통적인 기능을 제공하는 핵심공통기능 제공서버, 위치정보와 핵심기능을 수행하는 응용 솔루션, 이들 시스템과 상호작용하여 사용자 및 프로파일 관리 또는 고급 처리 및 모바일 트리거링(triggering)으로 구성된다.

LBS의 모니터링은 LBS 발신과정, LBS 호 연결 상태, 호 접속 상태에서의 LBS 시도과정에 의해 수행된다. 모니터링 수행은 모드에 따른 LBS 데이터를 원하는 횟수만큼 연속적으로 받을 수 있도록 설정하는 항목에 의해 수행되며, 위성 개수, 방위각, 시간, 위도, 경도, 속도, 높이, quality를 표시하는 항목에 의해 결정된다. 그리고 A-GPS, MS_BASED를 설정하는 항목에 의해 결정되며, LBS 발신 테스트, 호 연결상태, 접속상태에서의 테스트 결과에 의해 결정된다.

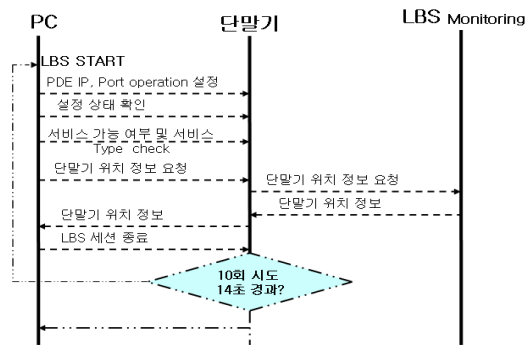
3.2.1 LBS 발신과정 모니터링

LBS 발신 모니터링은 PC에서 PDE IP, Port operation 설정, 설정상태 확인과정, 서비스가능 여부 및 서비스, 단말기 위치정보 요청 등을 단말기가 요청하면 단말기는 LBS와의

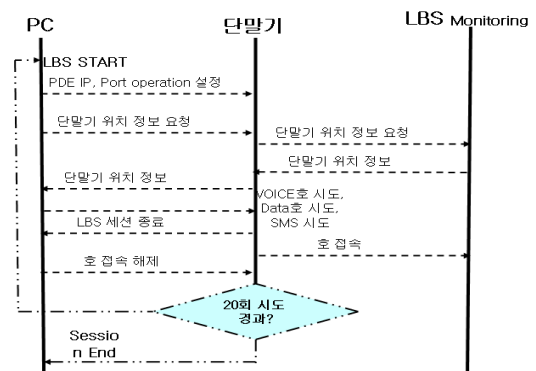
상호작용을 통해서 단말기를 거쳐 단말기 위치 요청 정보 및 단말기 위치정보를 모니터링하게 된다. 그림3에서 보듯이 LBS 시작에서 Session이 끝날 까지 걸리는 시간은 14초 이내로 설정하며, 반복적인 LBS 시도로 접속 성공률을 모니터링하게 된다.

3.2.2 LBS 호 연결 상태 모니터링

연결 상태 모니터링은 LBS 발신과정에서 연결상태를 모니터링하는 과정이며, 이때 LBS 호 신호를 위해 음성, 데이터, SMS로 구분하여 호의 신호상태를 모니터링하게 된다. 이 과정에서 LBS 호를 위한 Traffic이 종료되고 서비스를 요청을 신청한 서비스 사용자가 시도한 호가 연결되어 있는지를 확인하게 되며, 세션처리는 20회를 시도하여 정상 처리 여부를 확인하게 된다. 이와 같은 과정을 수행하는 모니터링 과정은 그림4와 같다.



[그림 3] LBS 호 시도 모니터링

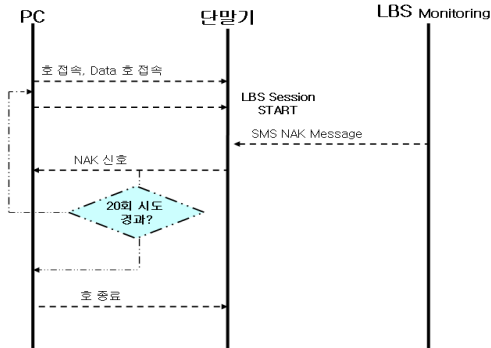


[그림 4] 연결상태 확인 모니터링

3.2.3 호 접속 상태에서의 LBS 모니터링

호 접속 상태에서의 LBS 모니터링 과정은 단말기에서 NAK 메시지가 수신 되는지를 모니터링하는 과정으로서 호 접속과 데이터 호 접속과정을 통해서 수신 상태를 모니터링

하게 된다. 이때 데이터 호를 처리하는 도중에 LBS 호가 Busy 상태인지를 파악하게 되며, LBS 세션을 20회 시도하여 모두 Busy 처리되는 지를 모니터링해야 한다. 만일 Busy 상태를 파악하여 LBS 세션이 20회를 초과하면 NAK 신호를 발생하여 호 처리를 중단하게 된다. 이와 같은 과정을 수행하는 모니터링 과정은 그림5와 같다.



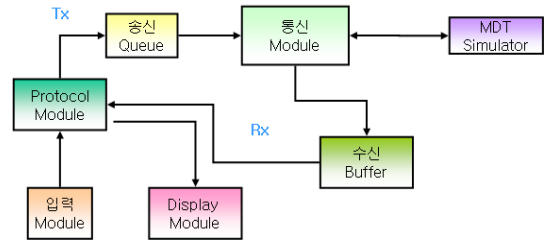
[그림 5] LBS Busy 모니터링

3.3 MDT Monitoring

MDT 모니터링은 차량의 무선 및 이동 서비스가 증가함에 따라 위치에 관계없이 사용자의 서비스 요구사항을 만족하기 위한 차량 단말상태를 모니터링하는 기능이다[11]. 이 기능은 주파수 공용 통신 시스템과 PCS 망을 이용하여 음성 및 문자 서비스를 모니터링하는 기능이며, 위치정보 전송, 차량 서비스 전송 등을 실시간으로 모니터링을 수행한다. MDT 모니터링은 외부 단말기와의 접속을 통하여 수행되며, MDT 테스트 상태를 모니터링 하는 기능을 수행한다. 이때 모니터링은 통신 모듈, Display 모듈, 입력 모듈로 구분되어 수행된다. 통신 모듈은 MDT 연동 규격에 맞도록 Mobile 단말기와 통신을 수행하여 이를 Auto_Test가 수행하도록 자동화는 기능이다. 즉 입력모듈에서 모니터링을 수행하게 될 명령어들이 차례대로 프로토콜 모듈로 전송되면 프로토콜 모듈에서 모니터링을 수행할 명령어들을 FIFO 큐에 넣는다. 이때 FIFO큐에 들어온 명령어들을 통신 모듈에서 순차적으로 탐색하여 단말기로 전송된다. 단말기에 전송된 명령어들은 통신 모듈에서 수신버퍼로 전송되며 전송된 명령어들은 순차적으로 프로토콜 모듈에서 분석된 후 그 결과를 Display 모듈에 의해서 디스플레이 된다. Display 모듈은 모니터링이 수행되는 결과들을 화면에 디스플레이 하는 기능으로서 전송되는 명령어 프로토콜과 수신 프로토콜을 분석하여 결과를 디스플레이 하게 된다.

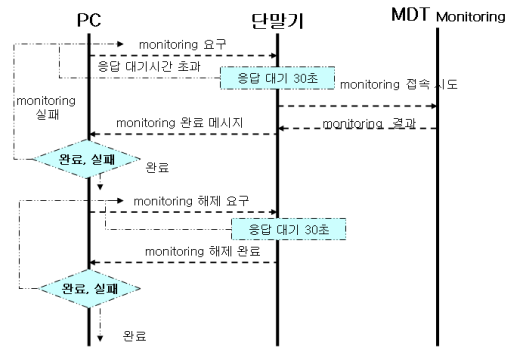
그리고 입력 모듈은 모니터링을 수행할 명령어들을 선택하거나 명령어 리스트를 입력하는 기능을 수행하게 되며 그

림6과 같다.



[그림 6] 입력모듈에 의한 MDT 모니터링

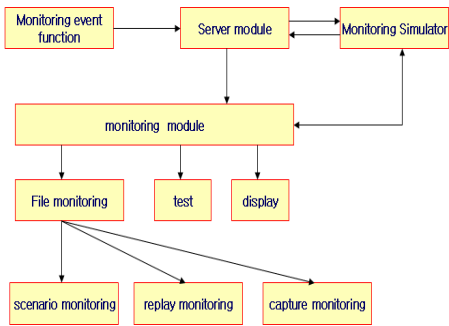
이와 같은 기능에 따라 MDT의 모니터링 상태 정보가 파악되게 되며 모니터링 상태 정보는 호 연결 요구를 한 후 응답 대기 시간에 따라 성공 여부가 결정된다. 이후 MDT 상태 확인은 DM_MONITORING_REQ 명령에 의해 수행되며 이와 같은 과정은 그림7과 같다.



[그림 7] MDT Monitoring 상태 확인

3.4 PDA Monitoring

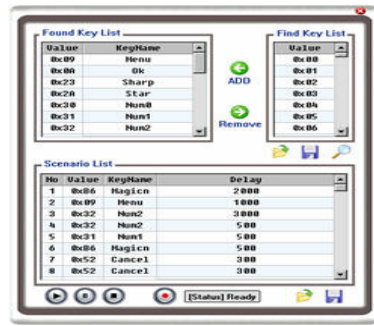
PDA Monitoring 은 모바일 단말기에 사용되는 PDA의 예외동작 또는 시스템 과부하 여부를 사전에 모니터링 하는 기능이다. PDA 모니터링은 모니터링을 수행할 기본사항과 키 동작 등을 순서화하여 예외 사항 발생 원인을 모니터링하기 쉽도록 지원하며 시나리오 실행시 최대한 빠른 속도로 예외처리 기능을 수행하도록 한다. 또한 LBS 모니터링과 MDT 모니터링의 부하와 오류 등을 관찰하여 모바일 성능 테스트의 이상여부를 파악하게 된다. 이러한 과정은 연산 오류, 메모리 오류, 그래픽 오류 등을 파악하여 모니터링 기능이 얼마나 잘 수행되는지를 분석하여 성능테스트 시간 단축 여부를 결정하게 되며, 결정 여부는 성능 시스템의 Driver를 통해서 체크하게 된다. 각 모니터링 체크 모듈은 모니터링 이벤트 함수와 서버모듈 및 모니터링 시뮬레이터와의 상호 작용을 통해서 수행되며, 수행과정은 그림8과 같다.



[그림 8] PDA monitoring

모니터링 이벤트 함수는 발생된 이벤트들을 감지하여 모니터링을 수행할 지 여부를 결정하는 함수로서 서버 모듈을 통해서 성능 테스트 명령 등을 전송하며, PDA 상태 정보를 모니터링 하게 된다. 모니터링 모듈은 scenario, replay, capture 등을 관리하는 기능이며 시나리오에 따라 이벤트가 수행되는지 그리고 이벤트들을 capture 해서 테스트가 가능한 이벤트인지를 모니터링하게 된다. 모니터링 상태가 수행되면 이들에 관한 결과를 테스트 하여 그 결과를 PDA 디스플레이를 통해서 동작 상태를 확인하게 된다.

된 상태를 모니터링 할 수 있도록 구성되어 있으며, 시나리오가 실행되면서 LBS 모니터링 상태가 화면에 표시된다. LBS 모니터링 실행시에는 초기 시나리오가 작동되어 단말기에 설정된 LBS 설정값을 읽어 올 수 있도록 하고 이에 대한 결과가 그림10에서처럼 화면에 디스플레이 된다. 이때 시나리오는 Trigger 설정 및 해지, Echo, Charge 등의 설정기능과 시나리오 작성, 편집, 저장 및 로그파일 생성 기능을 수행하고 시나리오에 의해 수행된 이벤트들은 그림9의 과정을 거쳐 모니터링 되며, 이에 대한 성능 관리 기능은 그림10과 같다.



[그림 9] 시나리오 리스트

4. 시스템 구현

4.1 구현환경

본 논문의 구현을 위해 InWireless상에서 EPM-K5500를 이용하였으며, MDT 모니터링 구현 모델은 LG-KB1800 모델을 이용하였다. 그리고 N-Zone, Kways 등 모바일 단말기에 제한된 시나리오 관리 모듈을 적용하여 필드 테스트 중 또는 차량 운행 중 성능관리 시스템 환경을 구성하였다. 성능 관리 시스템 구축을 위한 통신은 Diag Map으로 설정하고 단말기의 하드웨어 키와 소프트웨어 키를 사상시켜 시나리오를 작성하고 이를 실행하였다. LBS 모니터링 구현 모델은 SPH-9500을 이용하였으며 위치정보 서비스의 모니터링을 위해 자기 위치정보, 그룹위치정보 등을 이용하였다. 그리고 PPP 접속을 통하여 모바일 단말기의 Data 접속 상태를 유지하도록 하였으며 Data 접속 상태에서 지속적인 모니터링이 가능하도록 하였다. PDA 모니터링 구현모델은 CP-B361 모델을 이용하였으며, PC에서 원격제어가 가능하도록 하였다.

4.2 LBS 성능관리

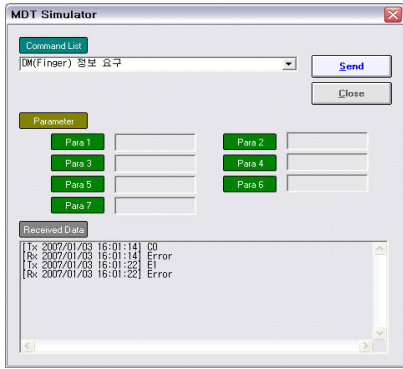
LBS 성능관리 구현 화면은 LBS가 시나리오에 의해 설정

[그림 10] LBS 성능 관리

4.3 MDT 성능관리

MDT 성능관리 구현 모델은 MDT 연동 규격에 맞는 Serial 통신에 관한 설정 변경이 가능하도록 구성하였으며, 통신 모듈의 경우 Serial 통신으로 전송되는 데이터를 FIFO 큐에 입력하고 수신되는 데이터를 수신 버퍼에 저장하여 통신을 수행한다. 그리고 Display 모듈에서 송/수신되는 데이터를 화면에 전송하고 전송된 프로토콜을 분석하여 결과를 화면에 Display하게 된다. 모니터링 상태를 테스트하기 위해 사용자가 직접 명령어를 선택하거나 명령어 리스트를 사용하도록 하였으며, MDT 단말 규격에 의한 통신을 수행하여 단말 모듈 상태를 확인하도록 하였다. MDT 규격에 따라 모바일 단말기의 상태정보 및 LBS 데이터 전송기능을 모니

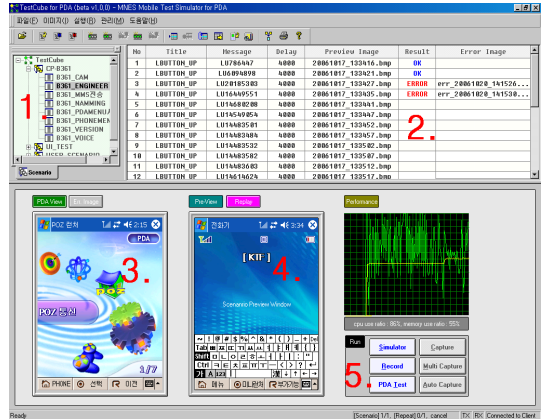
터링 하는 기능이 제공되며, 모바일 단말 규격에 의해 시나리오를 작성하여 MDT 규격에 정의되어 있는 단말기 동작이 수행되도록 구축하였다. 그리고 이동체에서 텔레서비스 식별자 값, 단말기 설정 값 등을 확인하고 변경 할 수 있도록 구축하였으며, 수신된 단문 메시지를 화면으로 보여줌으로써 모니터링이 자동으로 수행되는지 여부를 파악하게 하였다. 그림11에서 Command List는 이벤트들을 생성하지 않고 화면에서 모니터링 상태를 직접 파악하기 위해 사용되는 입력 값이다.



[그림 11] MDT 성능관리

4.4 PDA 성능관리

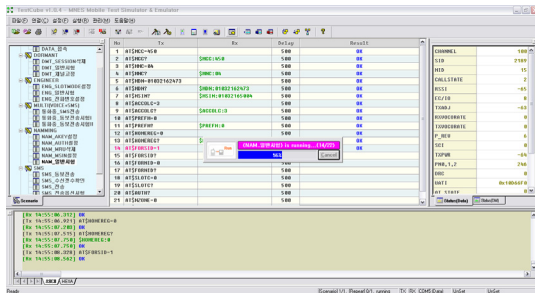
PDA 성능관리 모듈은 크게 설정 메뉴와 실행 메뉴로 구성되어 있으며, 실행 메뉴는 실제 프로그램이 구동되는 화면으로 화면에는 실제 프로세스의 동작 과정이 디스플레이되며, 결과 값을 출력하고 설정 메뉴에서는 프로그램이 실행되는 동작과 동작되는 시간, 항목 등과 설정, 변경, 해제 등의 모니터링 기능을 수행한다. 동작 내용을 모니터링하기 위해 버퍼 체크, CPU 부하 체크, 그래픽 에러 체크 등의 체크 기능과 LCD 점멸 등과 같은 기능 테스트가 수행되도록 하였으며, 테스트 기능 중 버퍼 체크는 메모리를 할당, 해제를 계속적으로 반복하면서 프로그램에서 할당된 데이터와 실제 데이터를 비교하도록 하였다. CPU 부하 체크는 특정 프로그램을 구동 시 실제 CPU가 얼마나 동작 중인지 %로 출력하게 되며 그래픽 에러 체크는 이미지를 포지션마다 썼다, 지웠다 하면서 그래픽 메모리에 오류의 발생 여부를 모니터링 하는 기능이다. 그리고 LCD back-light 점멸, HOTKEY 입력 테스트, LED 점멸 등의 동작을 수행하는 기능을 제공하며, 버튼 입력 시에는 어떤 버튼이 동작되었는지 표시 하는 기능이 화면에 표시된다. 시스템 정지를 제외한 나머지 상태에서 모니터링 상태에 대한 capture 기능이 가능하도록 하였으며 PDA 모니터링을 이용한 성능관리 그림12와 같다.



[그림 12] PDA 성능관리

5. 실험 평가

모바일 서비스의 단말 간(End-to-End)네트워크의 상태를 모니터링 하는 방법에는 라우터로부터 멀티캐스트 세션 참가자와 그 사이의 네트워크 상태를 모니터링 하는 방법, 라우터와 같이 다른 장비의 도움을 받지 않고 자신의 응용 프로그램에서 모니터링을 수행하는 방법, 다른 멀티캐스트 모니터링 도구를 이용하여 모니터링 하는 방법이 있다. 이러한 모니터링 방법은 주로 자원 사용량에 관한 모니터링 기법으로서 단말 사용 환경이 서로 다른 LBS, MDT 및 PDA 환경에서는 적합하지 않다. 이 장에서는 이와 같은 서비스 환경이 서로 다른 모바일 서비스의 성능을 알아보기 위하여 GPS(Global Positioning System)와 RTT(Round Trip Time) 상의 모니터링을 통한 시뮬레이션 성능을 비교 분석하였다. 시뮬레이션을 위한 simulator로는 inWirelee 상에서 수행되는 EPM-K5500을 사용하였으며 Emulator로는 LG-KB1800을 사용하였다. 그리고 LBS 서비스 성능평가를 위한 Simulator로는 SPH-9500을 사용하였으며, PDA Simulator로는 CP-B361을 사용하였다. 시뮬레이션 평가를 위한 패킷 상태는 패킷 상태 뷰를 통해 제한된 시스템의 모니터링과 GPS, RTT 상의 모니터링을 실시간으로 확인하여 수행되며, 상태 뷰에서 패킷 모니터링은 그림13과 같은 과정을 통하여 수행된다. 상태 뷰에서의 패킷 모니터링을 통한 시뮬레이션 항목은 143개의 시뮬레이션 패킷과 총 관리 시간을 157분 동안 수행하여 평가를 수행하였다.



[그림 13] 상태 뷰에서의 패킷 모니터링

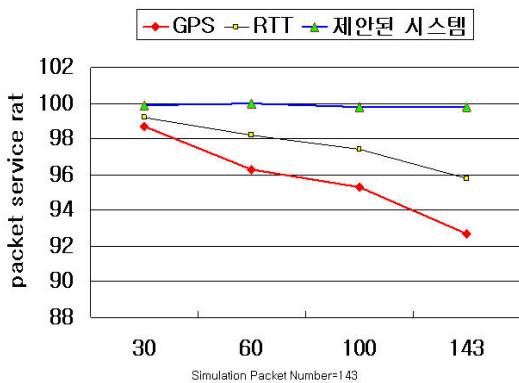
성능평가는 서비스 처리율과 패킷 오류 인식률에 따라 수행하였다. 서비스 처리율은 상태 뷰에서 관리된 패킷들을 서비스하기 위한 척도이며 서비스율(λ)는 다음과 같다.

$$\lambda = \sum_{c_i} \left[r_i \left\{ 1 - \prod_{\pi, j} (1 - \tau_{i,j}) \right\} \right]$$

여기서 τ 는 상태 뷰에서 패킷들이 서비스되는 값이다.

그리고 패킷 오류 인식율은 상태 뷰에서 인식된 패킷들이 서비스되지 않는 척도로서 그림13의에서 ok가 수행되지 않는 패킷들이다. 상태 뷰에서 발생된 오류들은 일반적으로 지터 지연(jitter delay), 패킷 간섭, 혼잡(congestion) 등에 의해 발생되며, 패킷 오류 인식률은 다음과 같다.

$$Packet_{error} = \sqrt{\frac{1}{NM} \sum_{x=0}^N \sum_{y=0}^M [View_{ok} - View_{failure}]^2}$$

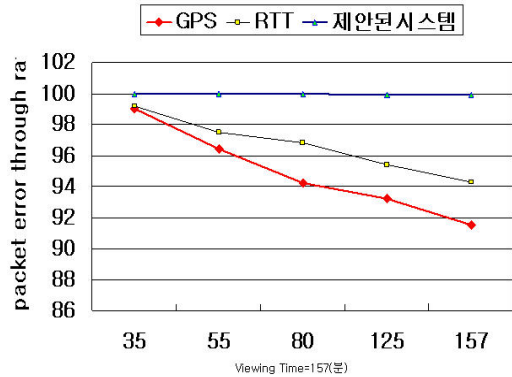


[그림 14] 패킷수에 따른 패킷 서비스율

그림14는 상태 뷰에서 패킷 모니터링을 통한 서비스 처리율이며, 그림에서 보듯이 제안된 시스템의 서비스 처리율이 GPS, RTT에 비해서 효율적임을 알 수 있다. 이러한 이유는 GPS를 이용한 모니터링 경우는 동일한 형태의

패킷 전송을 모니터링 있으며, RTT는 패킷을 수신할 때 서버의 상태에 따라 모니터링이 달라지기 때문이다.

그리고 그림15는 상태 뷰에서 157분 동안 수행된 패킷 오류 인식률이다. 시뮬레이션을 위해서 GPS와 RTT에서 수행된 패킷 전송은 시뮬레이션 조건이 같다는 가정 하에서 수행하였으며, 각각의 환경에서 157분 동안 수행된 패킷을 대상으로 하여 실험 평가하였다.



[그림 15] 시간에 따른 패킷 오류 인식율

그 결과 제안된 시스템에서는 패킷 오류 인식률을 거의 100% 정도 인식하는 데 반해서 RTT는 약 96%, 그리고 GPS는 94% 정도의 인식률을 보였다. 이 결과는 시뮬레이션 환경이 서로 다른 조건에서 수행되었지만 제안된 시스템은 모바일 서비스 관리를 LBS, MDT, PDA의 상태 뷰에 따라 실시간으로 모니터링 하기 때문이다. 따라서 제안된 시스템은 그림14와 그림15에서 보듯이 패킷 수와 시간에 관계없이 성능에 큰 차이가 없음을 알 수 있으며, 이에 따라 성능이 효율적으로 관리됨을 알 수 있다.

6. 결론

최근에 LBS, MDT 및 PDA 모바일 서비스가 일반화 되면서 이들 서비스에 대한 성능을 자동으로 관리하기 위한 여러 시스템들이 제안되고 있다. 본 논문에서는 모바일 사용자 필드에서 사용자와 단말기 간 그리고 단말기와 모바일 서버간의 서비스의 자동 관리 되도록 LBS 모니터링, MDT 모니터링, PDA 모니터링 기반의 성능 관리 시스템을 설계 구현하였다. 제안된 시스템은 시나리오 관리 모듈, MDT monitoring, LBS monitoring, PDA monitoring의 4개의 모듈로 구성되며 각 모듈의 상호작용을 통해서 서비스 성능관리가 수행된다. 성능 관리는

LBS, MDT, PDA의 성능 관리를 각각 독립적으로 수행할 수 있도록 하였으며 또한 통합 성능 관리가 수행되도록 하였다. 제안된 시스템의 성능을 알아보기 위하여 시뮬레이션 평가를 수행하였으며 시뮬레이션의 성능 평가는 상태 뷰에서의 패킷 모니터링을 통한 패킷 수에 따른 패킷 서비스율과 시간에 따른 패킷 오류 인식률을 비교 하였다. 그 결과 제안된 시스템의 성능이 GPS, RTT 시스템에 비해 성능이 우수함을 알 수 있었다.

참고문헌

[1] B. Housel and D. Lindquist, "Web Express : A System for Optimizing Web Browsing in a Wireless Environment," MOBICOM 96, pp. 108-116, Nov., 1996.

[2] CDG Document, Mobile IP Implementation Guidelines, V1.4, July, 1998

[3] ISO TC/211, 19132 Geographic Information - Location Based Services Possible Standards, <http://www.isotc211.org/scope.htm#19132>.

[4] LIF(Location Inter-Operability Forum), Statement Version 4, LIF.

[5] R. Floyd and B. Housel, "Mobile web Access using eNetwork Web Express," IEEE Personal Communications, pp. 47-52, 1998.

[6] TR45.6 Document, Wireless IP Network Architecture based on IETF Protocols, Jan. 1999

[7] Liang Xiao, Larry J. Greenstein, Narayan B. Mandayam, Shalini Periyalwar, "Distributed Measurements for Estimating and Updating Cellular System Performance," IEEE Trans. on Com., VOL. 56, pp. 991-998, 2008.

[8] Syed Ali Khayam, Shirish S. Karande, Muhammad Usman Ilyas, and Hayder Radha, "Header Detection to Improve Multimedia Quality Over Wireless Networks," IEEE Trans. on MULTIMEDIA, VOL. 9, pp. 377-385, 2007.

[9] 민경욱, 박중현, "LBS 플랫폼에서의 Web Services 구성 방법," 제20회 한국정보처리 학회 추계학술발표대회 논문집 제10권 제2호, pp. 689-692, 2003.

[10] 김용술, 김화성, "고품질 실시간 스트리밍 서비스 제공을 위한 네트워크 모니터링 기법," 한국통신학회 논문지, 01-11, vol. 31 No. 11B, pp. 969-978, 2006.

[11] B. Landfeldt, P. Sookavatana, A. Senviratne, "The Case for a Hybrid Passive/Active Network Monitoring Scheme in The Wireless Internet", in Proc. of IEEE international conference networks, 2002.

[12] B.B Lwekamp, "Combining active and passive network

measurement to build scalable monitoring systems on the grid," performance Evolution Review, 30(4): pp. 19-26, 2003.

정택원(Taegwon Jeong)

[정회원]



- 1981년 2월 : 서울대학교 대학원 전기공학과 (공학석사)
- 1991년 8월 : Univ. of Florida Dept of EE (Ph.D)
- 1983년 9월 ~ 1998년 8월 : ETRI 책임연구원
- 1998년 9월 ~ 현재 : 전북대학교 응용시스템 공학부 교수

<관심분야>
정보통신, 이동통신

이철한(Chul han Lee)

[정회원]



- 1983년 2월 : 원광대학교 전자공학과 졸업
- 1987년 8월 ~ 2001년 2월 : KT Networks 부장
- 2003년 2월 ~ 2004년 12월 : (주) 애니트론 부사장
- 2005년 1월 ~ 2009년 8월 : 현재 (주) 엠네스 대표

<관심분야>
Mobile 통신, Mobile 서비스, Mobile 성능관리

이종득(Chong Deuk Lee)

[정회원]



- 1989년 2월 : 전북대학교 대학원 전산통계학과 (공학석사)
- 1998년 2월 : 전북대학교 대학원 전산통계학과 (이학박사)
- 1992년 3월 ~ 2002년 2월 : 서남대학교 컴퓨터정보통신학과 교수
- 2002년 3월 ~ 현재 : 전북대학교 응용시스템 공학부 교수

<관심분야>
정보통신, 멀티미디어 통신, 무선통신, 모바일 성능평가