프록시 기반의 모바일 그리드 서비스 프레임워크 설계 및 구현☆

Design and Implementation of a Proxy-based Mobile Grid Services Framework

김 민 규* 이 봉 환** 박 시 용***
Min-Gyu Kim Bong-Hwan Lee Si-Yong Park

요 약

분산 컴퓨팅 환경에서 모바일 서로게이트시스템은 보안상의 문제로 인하여 널리 사용되고 있지 못하고 있다. 현재 가장 널리 사용되고 있는 그리드 미들웨어인 글로버스에서도 모바일 단말기를 인증하는 방법이 제공되고 있지 않아 그리드의 강력한 컴퓨팅 능력을 모바일 클라이언트까지 확장하는 것을 어렵게 하는 중요한 요인이 되고 있다. 본 논문에서는 PDA와 같이 컴퓨팅 파워가 약한 모바일 단말기와 그리드 호스트 사이에 인터페이스 역할을 하는 게이트웨이를 두어 모바일 클라이언트도 이동 중에 그리드 서비스를 제공받을 수 있도록 하는 모바일 그리드 서비스 프레임워크를 설계하고 구현하여 ECG 신호 처리를 위한 e-Healthcare 시스템에 적용하였다. 이 시스템은 이동 환경에서 환자의 생체 신호를 그리드 컴퓨팅으로 분석하여 원격에 있는 의사가 진단하는 모바일 헬스케어에 활용할 수 있다.

Abstract

In distributed computing paradigm, mobile surrogate systems have not gained wide acceptance because of security concerns that have not been suitably addressed yet. Currently even the widely used Grid middleware, Globus, does not have a host authentication mechanism, which makes it hard to provide high performance Grid computing capabilities to mobile clients. In this paper, a mobile Grid service framework is designed and implemented in order to provide Grid services to mobile host such as PDA. The surrogate host-based system allows mobile hosts to be provided with Grid services while moving. The implemented mobile Grid services system has been applied to mobile healthcare system which utilizes Grid computing for ECG signal processing. The presented system framework can be used as a secure enterprise mobile healthcare system for hospital physicians.

☞ Keyword : Distributed computing, Grid Security, Mobile Grid, e-Healthcare

1. 서 론

컴퓨터를 활용하기 위한 응용 및 시스템 개발은 일반적으로 안전한 중앙 집중적 환경에서 점차적으로 분산된 자원에 대한 데이터 공유 및 컴퓨팅시스템의 새로운 모드간의 상호 작용이 요구되는 형태로 발전하고 있다. 따라서 고성능 서버, 인공지능 네트워크, 네트워크 장비, 응용 서비스, 저장 시스템과 같은 고성능, 초고속, 대용량 자원간의 상호 연결을 필요로 하게 되었다. 이러한 추세에 편승하여 분산 응용 개발 및 활용에 대한 요구가 증가하고 이를 위한 미들웨어 및 응용의 개발이 다양하게 추진되고 있다. 대표적인 예로 지리적으로 분산되어 있는 고성능 노드들이나 호스트들을 연결하기 위한 미들웨어로 글로버스 툴킷(GT: Globus Toolkit) [1]이 출현하였다.

인터넷을 통해 사용자들이 정보를 공유할 수 있는 것과 마찬가지로 그리드 컴퓨팅을 사용하여 지역적으로 분산되어 있는 컴퓨터시스템들의 자

* 준 회 원 : (주)플랜티넷 연구원
kimmir@plantynet.com
** 종신회원 : 대전대학교 정보통신공학과 교수
blee@dju.ac.kr(교신저자)
*** 정 회 원 : 대전대학교 교육개발센터 교수
sypark@dju.ac.kr
[2007/01/02 투고 - 2007/01/08 심사 - 2007/01/18 심사완료]
☆ 본 연구는 정보통신부의 대학 IT연구센터 지원사업 (IITA-2006-(C1090-0603-0014)) 및 산업자원부의 지역혁신 인력양성사업 지원으로 수행되었음.

원들을 공유하며 협업 프로젝트를 수행할 수 있게 되었다. 그리드 컴퓨팅에서는 분산되어 있는 시스템들을 하나의 큰 가상 조직 (VO : Virtual Organization)으로 통합하여 단일 컴퓨터로는 효과적으로 처리할 수 없는 복잡한 문제들을 해결할 수 있다. 그리드에 연결된 시스템들은 같은 공간에 있을 수도 있고 전 세계에 걸쳐 분산될 수도 있으며, 다양한 운영체제를 갖는 하드웨어 플랫폼으로 구성될 수도 있다. 네트워크에 연결된 수많은 자원들의 안전한 사용을 위해 글로버스 툴킷에서는 그리드 시큐리티 인프라스트럭처(GSI : Grid Security Infrastructure)[2] 라는 개념을 도입하여 인증 받은 자원과 사용자만이 그리드에 안전하게 접근 할 수 있게 한다. GSI 개념에는 인증을 담당하는 SimpleCA[3], 온라인 인증서 저장소인 MyProxy[4], 단일인증(Single Sign-on)을 지원하기 위한 위임(Delegation), 원격사용자를 로컬 사용자로 매핑시키기 위한 grid-mapfile 등이 제공된다.

그러나 지금까지 GT는 인터넷에 연결된 슈퍼컴퓨터, 워크스테이션, 클러스터컴퓨터 그룹, 일반 데스크 탑 등 고정 IP를 가지고 유선으로 연결된 자원들만이 VO를 이루어 그리드 컴퓨팅을 할 수 있었다. 따라서 모바일 클라이언트에게 그리드 서비스를 제공하기 위해서는 모바일 단말기도 인증받았음을 확인할 수 있는 메커니즘이 필요하다. 이를 위해서는 SimpleCA에서 발행한 인증서와 관련된 여러 가지 정보를 데이터베이스에서 관리하며 한정된 사용자들에게만 인증서 정보가 저장된 데이터베이스와 연결된 게이트웨이를 거치게 함으로써 PDA나 핸드폰과 같은 모바일 단말기 사이에 데이터 교환과 그리드 서비스 이용이 가능해 질 수 있다. 본 논문에서는 모바일 단말기에서 SimpleCA에서 제공하는 패키지를 설치하지 않고 그리드 환경에 접근하도록 하기 위해 그리드 프록시 게이트웨이(GPG : Grid Proxy Gateway)를 설계하고 구현하였다. 또한 구현한 GPG를 e-Healthcare 시스템에 적용하여 그 효용성을 분석하였다.

본 논문의 2장에서는 그리드 보안 기반 구조인 GSI와 SimpleCA 및 MyProxy 구현에 대하여 기술하고 모바일 그리드 관련 연구에 대해서 알아본다. 3장은 GPG의 요구사항과 GPG 클라이언트 설계 과정 및 기능 구현에 대하여 기술한다. 4장에서는 GPG를 이용한 e-Healthcare 시스템에 대한 구현 결과를 보이고 5장에서 결론을 맺는다.

2. 관련 연구

2.1 그리드 보안 기반 구조

글로버스 툴킷에서 보안을 담당하는 부분은 GSI이며, 그리드 보안은 분산 자원들을 공유함에 따라 자연히 발생하는 가장 중요한 문제이다. 서로 다른 클러스터 환경 즉, 시스템 구조 및 네트워크 구조, 운영체제, 응용 라이브러리, 로컬 작업 관리자 등을 하나로 묶기 위해서는 단일화된 인증 구조가 필요하며, 이런 역할을 하는 것이 GSI이다. 그리드 보안 서비스가 필요한 이유는 다음과 같다.

- 계산 그리드(Computational Grid) 노드간의 안전한 통신
- 서로 다른 VO간의 단일화된 인증 구조의 제공
- 다양한 자원 혹은 사이트 간의 단일인증 제공

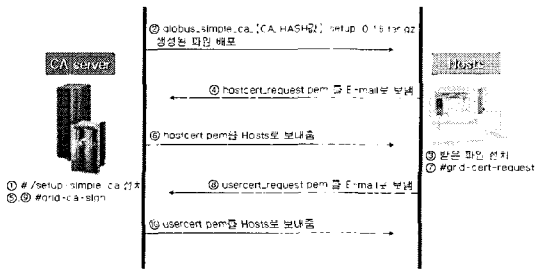
사용자의 입장에서는 안전하면서도 사용의 편리성을 요구할 것이고, 각 자원을 소유하고 관리하는 관리자의 입장에서는 자원이 그리드 환경에 개방되는 것이기 때문에 사용의 편리성 보다는 더 안전한 보안을 원한다. 이를 위해 글로버스(Globus) 프록시를 이용하여 GSI는 MyProxy와 Delegation을 이용한 단일인증 기능을 제공한다. 사용자는 그리드 환경에 단 한 번의 인증을 거침으로써 접근이 허용된 모든 자원들을 사용할 수 있고, 분산된 각 자원들에 대한 사용자 인증은 프록시가 대신 수행한다.

2.2 SimpleCA

GT4에서 제공하는 SimpleCA 패키지는 소규모 테스트베드에서 인증서버 (CA: Certificate Authority)를 구축하고자 하는 경우 사용된다. SimpleCA는 인증을 받은 노드들과 VO를 구축할 수 있는 방법을 제공한다. 하나의 VO에는 하나의 인증서버만이 존재하며, 인증서버에서 발행하는 인증서는 X.509v3 형식을 사용한다. 인증서버로부터 인증을 받으려는 노드는 인증서버가 SimpleCA를 설치한 후 생성되는 파일을 다운로드하여 설치하고 'grid-cert-request' 명령어를 이용하여 호스트 인증서와 사용자 인증을 받아야만 가상조직의 한 노드가 될 수 있다.

▪ 인증서 요청과 서명 방법

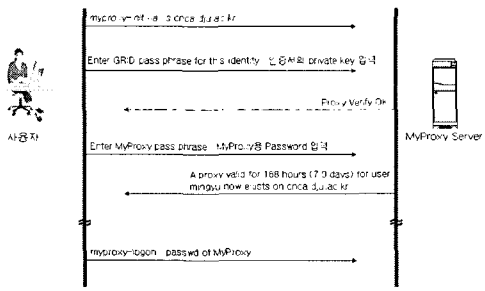
사용자가 인증서를 받으려면 사용자 노드에 글로버스 툴킷이 설치되어 있어야하고 인증을 받으려는 가상조직의 인증서버에서 배포하는 셋업 패키지를 다운받아 설치를 한다. 설치를 마친 사용자는 호스트 인증서와 사용자 인증서를 받아 정상적인 그리드 노드로서의 기능을 하게 된다. 호스트가 인증서버에게 인증서를 요청하고 설치하는 과정은 그림 2.1과 같다.



(그림 2.1) 인증서 설치 과정

2.3 MyProxy

MyProxy는 온라인 인증서 저장소로서 사용자들의 인증서를 안전하게 보관하는 역할을 한다. 이곳에 생성된 임시 인증서는 생성 후 일정기간 동안 사용할 수 있으며, MyProxy 서버로 로그인하여 인증서를 사용할 수 있는 기능을 제공한다. MyProxy 서버의 주요 기능 중 하나는 사용자 요청에 의한 인증서 저장 역할이다. MyProxy 서버에 인증서 저장 요청 시 사용자는 'myproxy-init' 명령어를 이용하여 인증서를 MyProxy 서버에 저장할 수 있다. 저장된 인증서는 사용자가 가상조직에 속한 다른 노드의 자원 즉, 프로세스 생성, 메모리 사용, 저장소 사용 등 자신이 누구인지에 대한 인증을 필요로 하는 경우에 저장되어 있는 인증서를 이용하여 사용자 인증을 대행한다. 사용자가 시스템 로그아웃 후 생성되어 있는 인증서를 재사용하려면 'myproxy-logon' 명령어를 이용하여 인증서를 재사용할 수 있다. 그림 2.2는 MyProxy 서버를 이용한 인증과정을 나타낸다.



(그림 2.2) MyProxy 서버를 이용한 인증 과정

2.4 모바일 그리드

최근 이동전화, PDA 등을 사용하는 모바일 사용자들이 급증하고 있으나 이들 모바일 장치들은 제한된 컴퓨팅 파워와 빈번한 연결 단절의 문제점을 내포하고 있다. 모바일 사용자들이 계산양이 많은 작업을 이동환경에서 실행하기 위해서는 사용자가 클라이언트로서 응용서버에 접속하고 실행 결과를 얻기 위해서는 서버에 지속적으로 접속된 상태로 남아 있어야 한다. 모바일 디바이스에게 모바일 그리드 컴퓨팅을 제공하기 위하여

다양한 메커니즘이 제안되고 있다. Millard 등은 모바일 디바이스를 위한 웹 프록시 기반의 그리드 클라이언트를 제안하여 글로버스 툴킷 3을 이용하여 구현하였다[5]. 그러나 이 시스템에서의 프록시는 단지 모바일 디바이스로부터의 요청을 그리드로 전송하는 기능만 가지고 있으며, 독립적인 프로세싱 기능은 전혀 제공하지 않는다. 모바일 OGSI.NET은 그리드 컴퓨팅을 제공하는 기존의 OGSI.NET을 모바일 디바이스들을 위해 확장한 것이다[5]. 모바일 OGSI.NET의 목적은 제한된 자원과 연결의 단절이라는 단점을 보완하기 위하여 모바일 사용자들을 위해 그리드 컴퓨팅의 장점을 제공하기 위한 것이다. 모바일 디바이스들을 위한 OGSI (Open Grid Services Infrastructure)의 구현은 실질적인 이동환경에서의 그리드 컴퓨팅을 제공하기 위한 시도라고 할 수 있다. 모바일 OGSI.NET은 OGSI 또는 WSRF (Web Services Resource Framework)에 기반을 둔 중요한 시도라고 할 수 있으나 현재 GT3과 호환 가능한 모바일 그리드 툴은 제공되고 있지 않다. 모바일 그리드를 위한 또 다른 시도는 계산양이 많거나 자원 집약적인 작업을 수행하기 위해 서로게이트(surrogate)를 이용하는 방법이다[6][7]. 제안된 three-tier 구조는 모바일 디바이스, 게이트웨이, 그리드 네트워크로 구성되어 있으며, 모바일 디바이스는 서비스 사용자가 되고 그리드 네트워크는 서비스 제공자가 되며, 게이트웨이는 모바일 디바이스로부터 작업 요청을 받아 이 작업을 그리드 네트워크로 전송하여 실행하도록 하는 브로커 역할을 담당한다. 이 제안은 현재 시스템 모델을 제시한 정도의 수준이며, 실질적인 구현은 제공하지 않고 있다. 더군다나 모바일 디바이스와 게이트웨이 및 그리드 호스트 간의 보안상의 문제에 대한 해결책을 제시하지 않고 있다. 모바일 디바이스들에게 그리드 컴퓨팅을 제공하기 위한 또 다른 방법은 모바일 에이전트를 이용하는 것이다 [8][9]. 모바일 에이전트는 클라이언트 시스템이 접속되어 있지 않은 상태에서도 작업을 실행할 수 있도록 한다. 작업 실행이 종료되고 난 후 클라이언트는 실행 결과를 검색하거나 타이밍이 중요한 경우 에이전트가 결과를 알려주게 할 수도 있다. 그러나 모바일 에이전트 접근법은 그리드 상에 있는 호스트들이 한 사용자나 하나의 기관에만 속하는 것이 아니기 때문에 보안 및 신뢰 문제를 야기할 수 있다. 여러 기관들의 참여로 인한 보안상의 문제를 해결하기 위하여 GSI는 종합적인 인증 및 권한검증 프레임워크를 제공하여 작업이 로컬 및 원격에 위치한 자원을 안전하게 엑세스할 수 있도록 한다. 따라서 모바일 그리드 컴퓨팅을 제공하기 위해서는 기존의 GSI 프레임워크와 모바일 에이전트 기술의 장점을 결합한 서로게이트 호스트 방식이 훌륭한 대안이 될 것이다. 이 방식은 GSI 기반의 보안 메커니즘을 모바일 도메인으로 확장하여 종단 사용자 디바이스들에게 컴퓨팅 능력을 제공하고 연결이 단절된 상태에서도 작업 실행을 제공한다. 동시에 이 방식은 모바일 에이전트 방식의 문제점인 복잡한 보안상의 문제를 제거할 수 있는 효과적이고 안전한 클라이언트-서버 모델이다. 따라서 본 논문에서는 GSI 보안 프레임워크와 서로게이트 호스트를 결합한 방식을 이용하여 모바일 그리드 서비스 프레임워크를 제안한다.

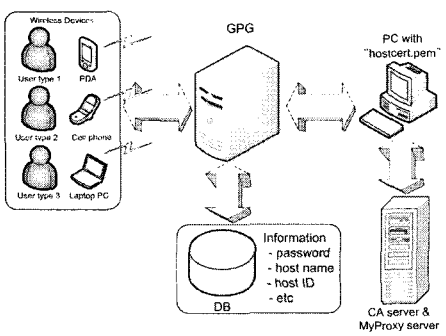
3. 프록시 기반의 모바일 그리드 인증 시스템 설계 및 구현

본 장에서는 본 연구팀의 기존 연구 결과를 확장하여 모바일 환경에서 그리드로의 접근을 제공하는 GPG의 설계 및 구현에 대하여 기술한다 [10][11]. GPG를 이용하여 입력장치가 없는 센서와 같은 소형 모바일 단말기로부터 전송되는 데이터를 그리드로 전송하는 방법과 입력장치가 있는 PDA와 같은 모바일 단말기로부터 전송되어온 명령어를 수행하는 과정을 설명한다. 시스템 구성에 사용된 장비의 사양은 모바일 노드와 센서의 역할을 하는 PDA의 경우 HP iPAQ rx3700을 사

용하였으며, 운영체제는 Microsoft사의 Windows Mobile 2003 Second Edition을 탑재하였다. JAVA 컴파일 버전은 1.3을 이용하였고 응용 구동을 위하여 IBM에서 제공하는 J9를 사용하였다. 한편, GPG의 CPU는 2.0GHz, 메모리는 512MB이고 운영체제는 Red Hat Linux 9.0 커널 2.4.20-8을 사용하고 JAVA 버전은 1.4.2_03-b02를 이용하여 구동하였다.

3.1 GPG 및 그리드 노드 설계

그리드 노드나 사용자가 인증을 받으려면 SimpleCA로부터 제공되는 셋업 패키지를 설치해야한다. 그러나 상용 모바일 장치의 경우 인증을 받기 위해 인증서버로부터 받은 패키지를 설치하고 인증서버로부터 인증서를 받기란 매우 어렵다. 인증서버로부터 인증을 받은 사용자가 모바일 장치에서 새로운 인증서를 받지 않고 기존의 인증서를 이용하여 그리드 네트워크에 쉽게 접근을 하도록 브로커 역할을 하는 것이 GPG이다. 따라서 기존에 인증서를 받은 사용자가 PDA를 이용하여 그리드 환경에 접근이 가능하도록 GPG를 설계하였다. 즉, GPG의 목적은 기존의 그리드를 모바일 그리드 환경까지 확장하여 언제 어디에서나 그리드 서비스를 제공받을 수 있도록 하는 것이다. 그림 3.1은 GPG의 기능 및 역할을 나타낸 것이다.



(그림 3.1) GPG의 기능 및 역할

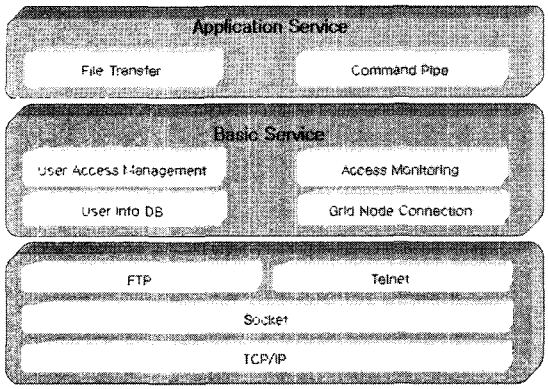
▪ GPG의 기능

GPG는 모바일 장치와 통신을 위한 브로커 역할을 하며 다음의 두 가지 경우가 있다.

- 입력장치가 있는 모바일 장치와의 통신
- 입력장치가 없는 모바일 장치와의 통신

위 두 경우 모두 장치 간 통신은 기본적으로 TCP/IP와 소켓을 이용하며, 입력장치가 없는 경우 자신이 어떤 장치인지 구분하기 위한 ID값은 JAVA 응용 코딩에서 구분한다. PDA에서 JAVA 응용 프로그램 구동을 위하여 IBM J9를 이용한다. JAVA는 원래 임베디드 시스템을 위해 개발되었고 특히, 최근 들어서는 다양한 하드웨어 이식성 때문에 임베디드 운영체제의 대안으로 부상하고 있다. J9에서는 JAVA에서 사용하는 FTP 클래스인 sun.net.ftp 클래스를 지원하지 않으므로 데이터 전송방법으로 소켓을 통하여 바이트 코드로 전송을 한다. 모바일 노드에서 전송하는 ID와 패스워드 관리와 자신의 인증서 위치를 나타내는 호스트 이름, 호스트 ID, 패스워드, 인증서 패스워드 등의 값을 저장하기 위해서 데이터베이스가 필요하다. GPG는 동시에 여러 접속자가 있을 수 있으므로 액세스 모니터링 기능이 필요하고 서버 관리측면에서 사용자 액세스 관리기능이 필요하다.

GPG와 그리드 노드간 통신을 위하여 역시 TCP/IP와 소켓이 필요하며 전송받은 데이터 전송을 위하여 FTP를 사용한다. 또한, 모바일 노드로부터 전송되어온 그리드 명령어를 실행하기 위한 'command-pipe' 기능이 필요하다. 그림 3.2는 GPG에서 요구되는 프로토콜과 정의된 기능 및 응용 서비스의 계층적 구조를 나타낸다. 또한, GPG는 모바일 장치로부터 전송되어 오는 명령어를 그리드 노드로 전송하기 위하여 텔넷(telnet) 기능이 필요하다.



(그림 3.2) GPG의 구조

▪ 그리드 노드 요구 사항

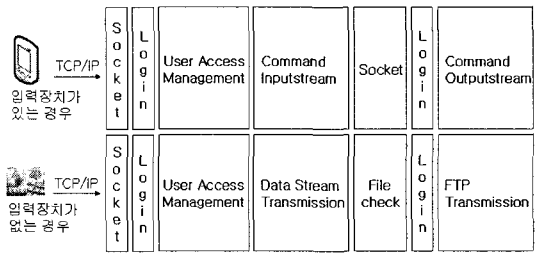
그리드 노드는 인증서버로부터 호스트 인증서와 사용자 인증서를 받아야하며 GPG로부터 전송되는 데이터를 저장하기 위하여 FTP 서버를 구동하고 있어야 한다. 또한, 명령어 수신을 위하여 텔넷 서비스를 구동하고 있어야한다. 모든 그리드 노드들은 루트 노드로부터 데이터를 전송받기 위하여 Gatekeeper 서비스와 GridFTP 서버가 구동중 이어야하며, XML 문서를 이용하여 ManagedJobFactoryService로 작업을 할당받는데 이를 위하여 그리드 컨테이너(container)를 실행하고 있는 상태여야 한다.

3.2 GPG 및 그리드 노드 구현

▪ GPG 기능 구현

GPG에서는 여러 사용자 접속을 가정하여 사용자 액세스 관리와 모니터링을 위하여 액세스별 스레드(thread)를 생성하여 다중 작업처리를 가능하게 하였다. 사용자별로 스레드를 관리하지 않으면 잘못된 노드로 명령어가 전송되거나 데이터 전송 오류가 생길 수 있다. 이렇게 함으로써 어떤 사용자가 어떤 명령어를 전송하는지 어떤 노드에서 데이터를 전송하고 있는지에 대한 모니터링과 사용자 액세스 모니터링이 가능해진다.

입력장치가 있는 경우 전송되는 스트링 값을 모두 받아 ID와 패스워드로 구분하고 데이터베이스 검색 결과와 비교하여 로그인을 하고 입력장치가 없는 경우 프로그램 실행 시 전송되는 ID와 패스워드를 데이터베이스 검색 결과와 비교하여 로그인할 수 있게 한다. 이 경우 로그인이 성공하면 곧바로 데이터 전송을 시작하게 되고 전송이 완료되면 GPG는 전송받은 데이터를 FTP를 이용하여 그리드 노드로 전송하게 된다. 그림 3.3은 모바일 장치별 데이터 처리 과정을 나타낸 것이다.



(그림 3.3) 모바일 장치별 처리 과정

▪ 그리드 노드 기능 구현

그리드 노드의 기능을 구현하기 위해서는 리눅스에서 제공하는 Telnet을 실행하고 추가적으로 GridFTP 및 Gatekeeper 서비스를 실행해야 한다. GridFTP 서비스는 그리드 환경에서 분산된 데이터를 안전하고 신속하게 그리드 노드로 전송하거나 저장하는 기능을 제공하며, 빠른 전송을 위하여 다중 포트를 이용하한 병렬전송 기술을 사용한다[12]. Gatekeeper는 원격지의 그리드 노드로부터 그리드 명령어가 전송되어 올 때 반드시 거치는 관문이다. Gatekeeper 서비스가 구동되고 있지 않으면 보안관련 오류가 발생하며 그리드 노드로서 역할을 올바르게 수행할 수 없다. 'globusrun-ws' 명령어를 이용하여 XML 파일의 형식으로 전달되는 작업파일을 받기 위하여 컨테이너의 ManagedJob FactoryService를 이용한다. 따라서 작업을 수행할 모든 그리드 노드는 그리드 컨테이

너를 실행하고 있어야 한다.

- grid-mapfile 설정

grid-mapfile은 다중 사용자가 자신의 인증서로 접속을 시도할 때 로컬영역에서 어떤 사용자로 매핑을 할 것인지를 정하는 설정 파일이다. 이 파일 등록 시 'grid-mapfile-add-entry' 명령어를 사용하며, '-dn' 옵션에 삽입하는 내용은 인증서의 subject 부분이고 '-ln' 옵션에 삽입되는 내용이 로컬영역에서 사용될 ID이다. 이렇게 하여 로컬영역에서 사용되는 ID에 대한 권한을 얻게 된다. 인증서버로부터 받은 인증서는 ~/.globus/ 하위에 있어야만 하며, 인증서가 다른 위치에 있을 경우 인증서를 찾지 못하고 보안 오류가 발생한다. 따라서 다른 host로부터 인증 받은 사용자들을 등록하기 위해서는 임의의 ID를 만들어 사용하면 된다. mapfile에 등록된 사용자는 매핑된 ID 권한만큼의 권한을 얻게 된다.

- 컨테이너 인증서 생성

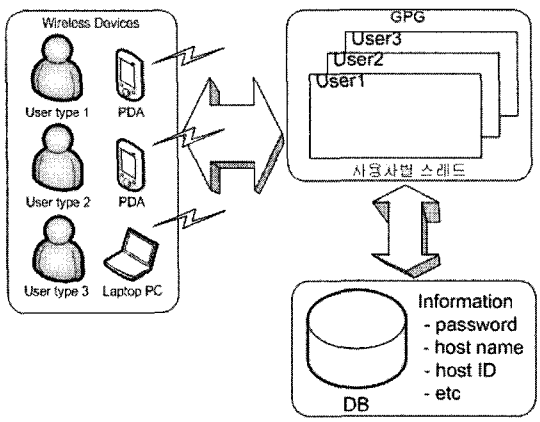
그리드 컨테이너 실행 시 필요한 마지막 요건으로 컨테이너 인증서가 필요하다. 호스트 인증서를 받았다면 컨테이너 인증서는 별도로 인증 받을 필요 없이 호스트 인증서를 컨테이너 인증서로 복사하고 사용자 권한을 글로버스 툴킷 관리자로 전환해주면 된다.

3.3 모바일 그리드 클라이언트 설계

우선 사용자는 기존에 인증서를 받은 사용자라고 가정한다. 사용자는 임의의 인증서버로부터 인증서를 받아 자신의 서버 ~/.globus/ 하위에 인증서가 보관되어 있어야한다. 또한 사용자는 GPG에 접속을 위하여 사전에 GPG 데이터베이스에 필요한 개인 정보를 저장해 놓아야 한다. 그리드 서비스를 제공받기 위해 모바일 장치 사용자는 모바일 그리드 클라이언트 응용을 실행한다. 즉, GPG에 접근하기 위하여 ID와 패스워드를 입력하면 데이터베이스로부터 정보를 검색하여 사용자 로그인 여부를 판단한다. 로그인이 성공하면 GPG는 인터넷상에 연결되어 있는 인증서가 저장되어 있는 호스트로 접속을 하게 된다. 이렇게 하면 모바일 장치에서 호스트를 원격 제어할 수 있는 상태가 된다. 이때 'myproxy-init' 명령어를 전송하여 MyProxy 서버에 인증서를 생성한 후 그리드에 접근을 한다.

3.4 모바일 그리드 클라이언트 구현

GPG의 첫 번째 역할은 모바일 디바이스로부터 사용자의 접속을 허용하는 것이다. 각 사용자별로 변수의 공유를 막고 사용자별 처리를 위하여 GPG에서는 새로운 사용자 별로 스레드를 생성하여 사용자의 요청을 처리한다. 새로운 접속이 있을 때 마다 사전에 등록된 DB를 검색하여 사용자 정보를 확인하고 로그인을 허용한다. 그림 3.4는 이러한 GPG의 동작 과정을 나타낸 것이다.



(그림 3.4) GPG 내부 동작 과정

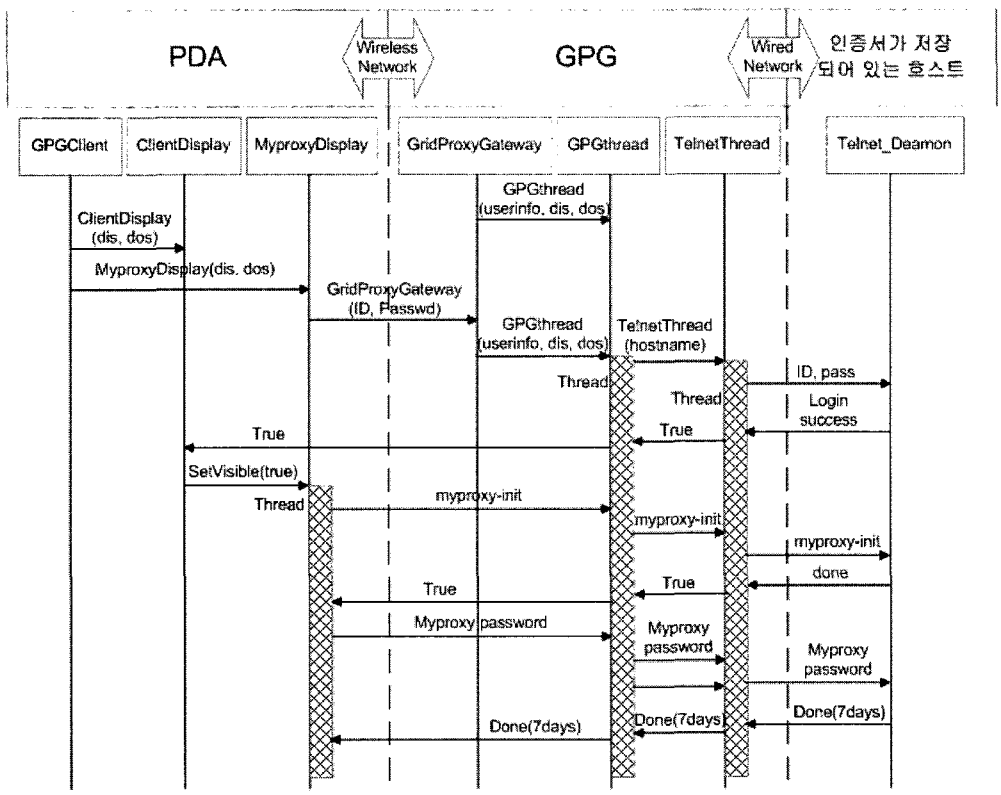
사용자별로 생성된 스레드는 모바일 디바이스와 인증서가 저장된 호스트를 중계하는 브로커 역할을 하게 된다. 그림 3.5는 GPG를 통한 여러 가지 기능 중 PDA, GPG 및 인증서를 가지고 있는 그리드 노드 사이에 인증을 위한 일련의 과정

을 나타내고 있다. 먼저 사용자는 PDA 상에 구현된 모바일 그리드 클라이언트를 구동하여 GPG에 접속하고 ID와 패스워드 정보를 입력한다. GPG는 사전에 등록된 DB를 검색하여 사용자 정보를 확인하고 합법적인 사용자인 경우 인증서가 저장되어 있는 호스트로 사용자 정보를 전송한다. 호스트는 사용자 정보를 확인하고 확인 응답을 GPG를 거쳐 PDA로 전송한다. PDA 사용자는 GPG로 하여금 'myproxy-init' 명령어를 구동하도록 하여 프록시 인증 요청을 인증서가 저장되어 있는 호스트로 보내고 이 호스트는 MyProxy 서버로 인증 요청을 전송하고 그 결과를 GPG를 경유하여 이동 단말인 PDA로 보낸다.

4. 모바일 그리드 게이트웨이 기반 e-Healthcare 시스템

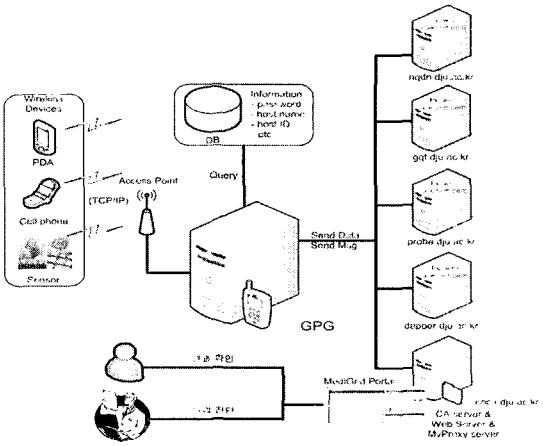
4.1 GPG 기반 e-Healthcare 시스템 모델

GPG는 다양한 응용에 적용할 수 있으며, 본 연구에서는 이를 e-Healthcare 시스템에 적용하였다. 모바일 그리드 게이트웨이 기반의 e-Healthcare 시스템 구성도는 그림 4.1과 같다. e-Healthcare 시스템 테스트베드는 모바일 호스트, GPG, 웹 서버 및 네 대의 그리드 노드들로 구성된다. 사용자는 e-Healthcare 시스템에 접근하기 전에 GPG에 접근하여 사용자 인증을 받은 다음 환자의 ECG(Electrocardiogram) 데이터를 모바일 노드에서 GPG로 전송한다. GPG가 수신한 ECG 데이터를 웹 서버 (인증서버, MyProxy 서버)로 전송하면 웹 서버는 네 대의 그리드 노드에 작업을 분배하여 처리하고 그 결과를 웹 서버로 되돌려 준다. ECG 데이터 처리는 PhysioNet[13]에서 제공



(그림 3.5) 클라이언트, GPG 및 인증서 호스트간 인증을 위한 상호작용

하는 특징 파리미터 추출 알고리즘을 사용하여 P, Q, R, S, T, PR, ST 세그먼트 등을 추출한다. 처리된 결과는 웹 서버에 저장되고 원격에 있는 의사에게 통보되어 원격 진료를 할 수 있게 한다. 원격의 의사의 진료 소견 및 ECG 데이터의 처리 결과는 이동 중에 있는 환자의 PDA로 통보되어 그 결과를 알 수 있게 한다.



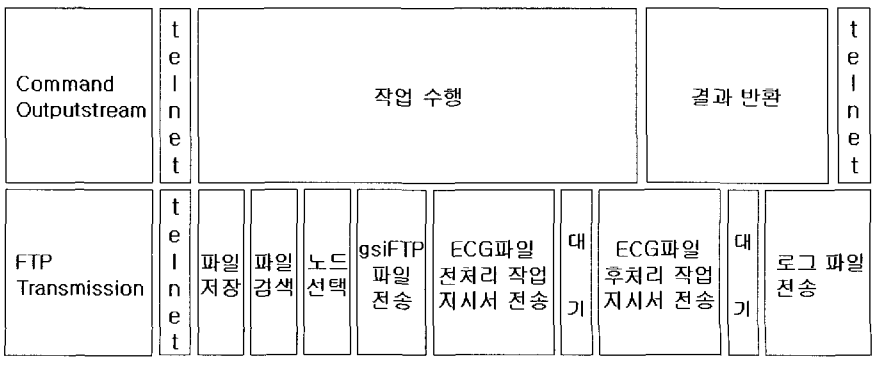
(그림 4.1) GPG 기반의 e-Healthcare 시스템

4.2 PhysioNet

본 실험에서 사용한 ECG 데이터는 미국 국립 보건원 (National Institutes of Healthcare: NIH)의 지원으로 하버드 의과대학과 MIT에서 공동으로 구축한 실제 환자 및 정상인의 심장 관련 데이터이다. PhysioNet은 웹 기반의 클라이언트-서버 구조로 이루어진 연구, 개발용의 생체 및 의료정보 네트워크 데이터베이스를 포함하고 있으며, PhysioBank와 PhysioToolkit과 함께 제공된다. PhysioBank는 건강한 상태의 정상인 또는 질병을 가진 환자로부터 추출한 데이터를 특성화하여 저장해놓은 데이터베이스로 현재 30GB의 PhysioBank Archive 데이터를 자유롭게 다운로드 할 수 있으며, 이 데이터는 Chart-O-Matic을 사용하여 웹 브라우저를 통해 볼 수 있고, 이메일을 통해 전송하여 사용할 수도 있다. PhysioToolKit은 WFDB (WaveFormDataBase) 형태의 소프트웨어 라이브러리이며, PhysioBank의 데이터를 처리, 분석, 디스플레이 하기 위해 PhysioNet에서 제공하는 소프트웨어들의 모음이다. 모든 소프트웨어들은 소스 형태로 제공되어 사용자가 필요에 따라 변경하여 사용할 수 있고, 새로운 알고리즘에 대한 연구 및 검증이 가능하다.

4.3 모바일 e-Healthcare 시스템 구현

모바일 그리드 게이트웨이 기반의 e-Healthcare 시스템을 구현하기 위해서는 GPG로부터 파일 전송 여부를 실시간으로 모니터링하기 위한 파일 검색 기능이 필요하다. 또한, 전송된 파일을 처리하기 위하여 globusrun-ws 명령어를 이용하여 XML 형식의 작업지시서를 그리드 노드로 전송하기 위한 기능도 추가적으로 필요하다.



(그림 4.2) 파일 검색과 처리 순서도

4.3.1 파일 검색

파일 검색 프로그램은 JAVA로 구현했으며 /home /globus/ecg/ 하위에 새롭게 생성되는 파일의 이름을 모니터링하는 것이 주목적이다. /home/globus/ecg/는 센서로부터 FTP를 통해 전송받은 ECG 파일이 저장되는 곳이기도 하다. 즉, 센서로부터 전송되어 오는 새로운 파일의 이름을 검색하는 것이며, 파일 중 ECG 파일이 생성되면 해당 파일의 분석을 요청하게 된다. ECG 파일 분석의 결과로써 로그 파일이 생성되면 이 결과들을 웹상에서 확인할 수 있도록 Tomcat 웹 서버로 전송한다. 그림 4.2는 파일 검색 및 처리에 대한 순서도를 나타낸다.

4.3.2 작업 스케줄링

모바일 그리드 게이트웨이 기반의 e-Healthcare 시스템에서 사용한 작업 스케줄링 방법은 라운드 로빈 방식이다. 사용 가능한 노드의 IP 주소를 파

```
<?xml version="1.0" encoding="UTF-8"?>
<multiJob xmlns:gram="http://www.globus.org/namespaces/2004/10/gram/job"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <factoryEndpoint>
        <wsa:Address>
https://203.237.141.195:8443/wsrf/services/ManagedJobFactoryService
        </wsa:Address>
        <wsa:ReferenceProperties>
            <gram:ResourceID>Multi</gram:ResourceID>
        </wsa:ReferenceProperties>
    </factoryEndpoint>
    <directory>${GLOBUS_USER_HOME}/mpich/ecg</directory>
    <count>1</count>
    <job>
    <factoryEndpoint>
    <wsa:Address>
https://203.237.141.195:8443/wsrf/services/ManagedJobFactoryService        </wsa:Address>
            <wsa:ReferenceProperties>
                <gram:ResourceID>Fork</gram:ResourceID>
            </wsa:ReferenceProperties>
        </factoryEndpoint>
<executable>${GLOBUS_USER_HOME}/mpich/ecg/ecgpuwave</executable>
        <argument>-r</argument>
         <argument>103</argument>
        <argument>-a</argument>
        <argument>ann</argument>
     <stdout>${GLOBUS_USER_HOME}/mpich/ecg/stdout.p1</stdout>
     <stderr>${GLOBUS_USER_HOME}/mpich/ecg/stderr.p1</stderr>
    <count>1</count>
    </job>
</multiJob>
```

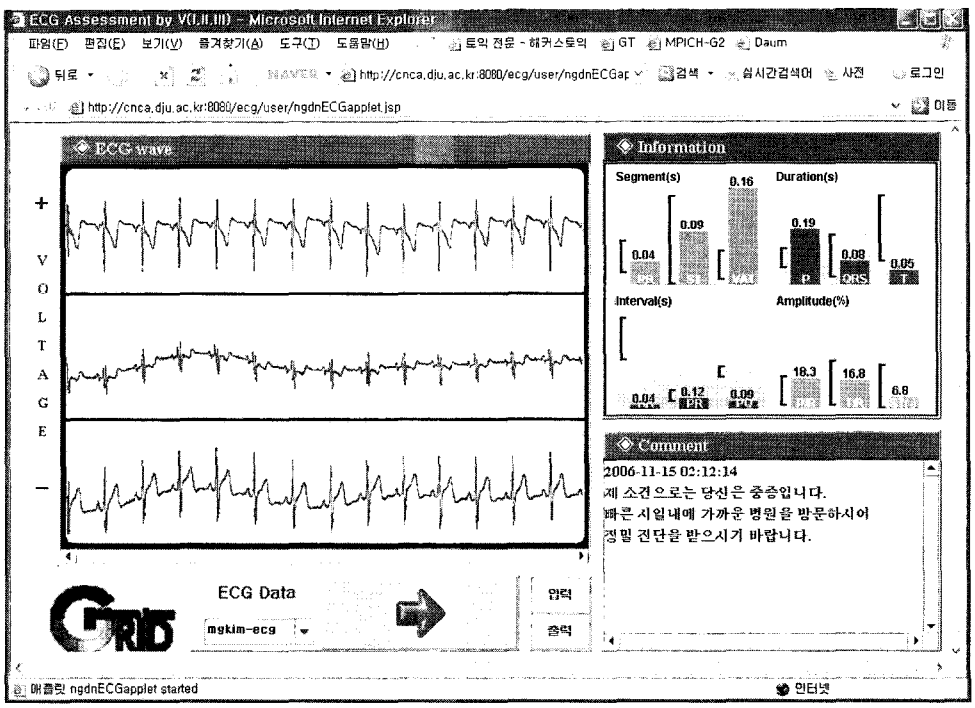
(그림 4.3) 전처리 작업 지시서

일로부터 입력받아 선입선출(FIFO) 방식으로 작업을 처리한다. 그림 4.3에 ECG 신호 전처리 과정이 자세히 기술되어 있다. <wsa:Address> 태그의 인자 값을 살펴보면 작업을 수행할 노드의 IP 정보가 기입되어 있음을 알 수 있고 포트 정보와 함께 컨테이너의 어떤 서비스를 이용하는 지에 대한 정보도 포함되어 있다. <directory> 태그의 내용은 다중 작업 실행 시 권한을 갖는 디렉토리를 명시하며, <count> 태그는 자신의 작업을 몇 번 실행 할 것인지에 대한 횟수를 나타낸다. 이렇게 하여 선언부가 끝나게 되고 <job> 태그는 작업의 내용을 기록하기 위한 것이다. 위에서 설명한 바와 마찬가지로 <wsa:Address> 태그는 작업 노드 IP 주소, 포트 번호, 서비스명을 명시하고 있다. <gram:ResourceID> 태그는 다중 작업을 시키기 위한 것이므로 XML 문서를 받는 노드에 프로세스를 생성하겠다는 의미가 된다. <executable>에는 실제로 실행될 명령어가 들어가게 되고 <argument>에는 그에 필요한 옵션을 기입할 수 있다. <stdout>은 정상적인 <executable>이 실행되었을 때 결과를 출력하는 파일명을 지정하기 위한 것이고 <stderr>은 <executable>가 실행 중 발생 오류를 저장하기 위한 것이다.

XML 파일은 총 두 종류로 구분되어 시스템에서 실행되고 있으며 그 중 한 가지는 '파일명 ecgpuwave.xml'로서 ECG 파형 분석을 위한 전처리 명령어를 포함하고 있다. 다른 하나는 '파일명rdann.xml'로서 전처리 결과를 이용하여 ECG 특징 파리미터들을 추출하기 위한 명령어를 포함하고 있다.

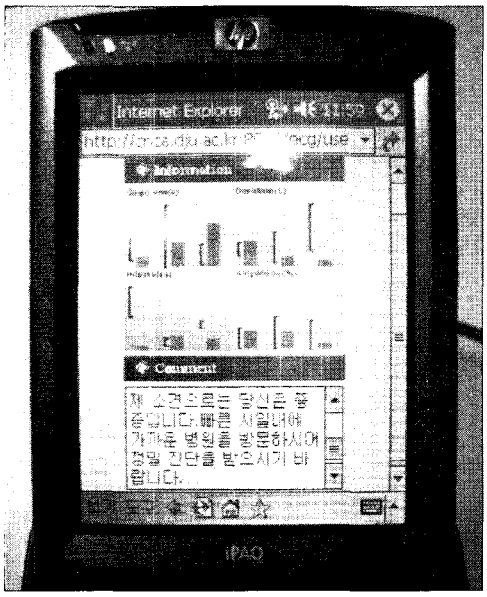
4.4 실행결과

검색 결과 파일을 웹서버로 전송하는 것은 원격지의 의사가 ECG 데이터를 보고 진료를 보다 쉽게 하기위해서이며, 환자 또한 의사의 진찰내용



(그림 4.4) ECG 신호 처리결과 및 의사 진단 소견 화면

을 웹을 통하여 PDA에서 손쉽게 열람할 수 있게 해준다. 그림 4.4는 ECG 신호 처리결과 및 의사 진단 소견 화면으로써 <ECG wave>는 환자가 보낸 ECG 파형을 그래프로 표시하는 부분이고 <Information>은 입력 ECG 신호의 분석 결과로써 ecgpuwave 프로그램의 결과인 로그 파일을 이용하여 Segments, Duration, Interval, Amplitude 등의 값을 연산하여 막대그래프로 표시한 것이다. 환자는 막대 그래프의 허용 구간과 처리 결과 값을 비교하여 자신의 ECG 신호로부터 추출한 각 파라미터들이 어느 구간에 위치하고 있는지를 쉽게 확인할 수 있다. 우측 하단의 <Comment>는 ECG 파형과 Information의 정보를 토대로 의사가 진단 소견을 입력하는 부분이다. 한편, 그림 4.5는 환자가 PDA로 자신의 ECG 신호에 대한 의사의 진단 결과를 확인하는 화면이다.



(그림 4.5) PDA 접속 진단 확인 화면

5. 결 론

본 논문에서는 기존의 그리드 보안 서비스 기술을 확장하여 이동 환경에서도 그리드 서비스가 가능한 모바일 그리드 서비스 프레임워크를 설계하고 구현하였다. 또한, 구현한 모바일 그리드 서비스 프레임워크의 효용성을 검증하기 위하여 이를 e-Healthcare 시스템에 적용하였다. 모바일 그리드 보안은 기존에 유선 환경에서 제공되고 있는 GSI를 이동 환경으로 확장한 것으로 프록시 게이트웨이(GPG)를 기반으로 설계하였다. 즉, 모바일 디바이스들은 그리드 네트워크 상에 위치하고 있는 그리드 호스트를 액세스하여 자신의 프록시 인증서를 제공받아 안전하게 원격 작업을 실행할 수 있도록 구현하였다. 구현한 모바일 그리드 인증시스템은 PDA, 인증서버 및 네 대의 노드로 구성된 그리드 네트워크 테스트베드 상에서 정상적으로 동작함을 확인하였다.

GPG를 이용한 모바일 그리드 시스템은 이동 중에 있는 환자가 PDA을 이용하여 자신의 ECG 신호를 그리드 네트워크로 전송하여 처리결과 및 의사의 소견을 자신의 PDA로 확인할 수 있도록 구현되어 있다. 즉, PDA에서 획득된 ECG 신호는 GPG로 전송되어 그리드 상에 있는 노드에서 심혈관계 신호처리 알고리즘을 사용하여 처리된다. 또한, 웹 서비스 기반의 그리드 컴퓨팅을 이용하여 처리된 특징 파리미터 결과들은 파형과 함께 원격지의 의사에게 제공되고 의사는 이 정보를 토대로 진단을 내려 웹 서버에 저장한다. 저장된 특징 파리미터 그래프와 의사의 소견은 환자의 PDA로 전송되어 환자가 필요한 조치를 취할 수 있도록 설계하였다. 모바일 e-Healthcare 시스템에서 사용한 그리드 시스템의 성능을 분석하기 위하여 사용된 데이터는 PhysioNet에서 제공하는 실제 병증별 ECG 데이터이며, 이 시스템은 이동 환경에서 환자의 생체 신호를 그리드 컴퓨팅으로 분석하여 원격에 있는 의사가 진단하는 모바일 헬스케어에 활용할 수 있다. 향후 연구 내용으로 ECG 뿐 아니라 EEG 등 다양한 생체 신호에 대한 e-Healthcare 시스템의 개발이 요구된다 하겠다.

참 고 문 헌

[1] http://www.globus.org/toolkit/.

[2] http://www.globus.org/toolkit/docs/3.2/gsi/.

[3] http://www.globus.org/grid_software/security/simple-ca.php.

[4] http://www.globus.org/grid_software/security/myproxy.php.

[5] David E.Millard, ArounaWoukeu, and Feng Tao. Experience with writing grid clients for mobile devices. In 1st International ELeGI Conference, 2005.

[6] D.Chu and M.Humphrey. Mobile ogsi.net: Grid computing on mobile devices. In 5th IEEE/ACM InternationalWorkshop on Grid Computing, November 2004.

[7] Sungyoung Lee Taewoong Jeon Saad Liaquat Kiani, Maria Riza and Hagbae Kim. Grid access middleware for handheld devices. In Europe Grid Computing 2005, pages LNCS 3470 1002 - -1011, 2005.

[8] Ali Sajjad Sungyoung Lee Hassan Jameel,Umar Kalim and Taewoong Jeon. Mobile-to-grid middleware: Bridging the gap between mobile and grid environments. In Europe Grid Computing 2005, pages LNCS 3470 932 - -941, 2005.

[9] H. K. Neo, Q. P. Lin and K. M. Liew, "A Grid-Based Mobile Agent Collaborative Virtual Environment," In Proc. of International Conference on Cyberworlds (CW'05), pages 335-339 , Nov. 23-25. 2005.

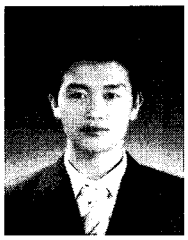
[10] 김민규, 김정수, 조현숙, 이봉환, "프록시 게이트웨이를 이용한 모바일 그리드 보안시스템," 2006 춘계학술발표대회 논문집 제7권 1호, pp.315-318, 한국인터넷정보학회, 성균관대학교, 2006.4.28-4.29.

[11] Bong-Hwan Lee, Min-Kyu Kim, Hyun-Suk Cho, Se-Yul Lee, Chan-Hyun Youn, and Jose Fortes, "A Secure Surrogate Host System for Mobile Grid Services," In Proc. of WISA 2006, pp.543-550, Jesu, Korea, August 28-30, 2006.

[12] 김영신, 허의남, 김일중, 황준, "병렬 연결 간의 트래픽 간섭 현상 분석 및 대역폭 예측" 인터넷 정보학회 논문지 Vol.7, No.1. FEB 2006. 2월호.

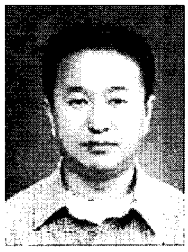
[13] http://www.physionet.org.

◐ 저 자 소 개 ◑



김 민 규 (Min-Gyu Kim)
2005년 대전대학교 정보통신공학과 졸업(학사)
2007년 대전대학교 대학원 정보통신공학과 졸업(석사)
2007~현재 (주)플랜티넷 연구원
관심분야 : 그리드컴퓨팅, 웹서비스, 네트워크보안 등
E-mail : kimmir@plantynet.com

이 봉 환 (Bong-Hwan Lee)
1985년 서강대학교 전자공학과 졸업(학사)
1987년 연세대학교 대학원 전자공학과 졸업(석사)
1993년 Tesax A&M 대학교 대학원 전기공학과 졸업(박사)
1995~현재 대전대학교 정보통신공학과 교수
관심분야 : 그리드컴퓨팅, 웹서비스, 네트워크보안 등
E-mail : blee@dju.ac.kr



박 시 용 (Si-Yong Park)
1997년 경성대학교 전자계산학과 졸업(학사)
2001년 부산대학교 대학원 멀티미디어학과 졸업(석사)
2005년 부산대학교 대학원 전자계산학과 졸업(박사)
2006~현재 대전대학교 교육개발센터 교수
관심분야 : 모바일 인터넷, 멀티미디어 스트리밍, 임베디드 시스템 등
E-mail : sypark@dju.ac.kr