

이동 환자 생체신호의 실시간 전달을 위한 오버레이 네트워크 기반 자율군집형 미들웨어 플랫폼

강 호 영*, 정 설 영*, 안 철 수**, 박 유 진*, 강 순 주°

Self-Organizing Middleware Platform Based on Overlay Network for Real-Time Transmission of Mobile Patients Vital Signal Stream

Ho-Young Kang*, Seol-Young Jeong*, Cheol-Soo Ahn**, Yu-Jin Park*, Soon-Ju Kang°

요 약

이동환자의 생체진단신호 원격전달을 위해서는 환자 및 감시자의 이동성, 환자의 이상징후 감지기능과 관련 컴퓨팅 자원들의 자율군집성 동작 서비스 바인딩 기능이 필수적으로 요구된다. 기존의 연구는 이동 환자 생체 신호 전달을 위해 중앙 집중화된 방식으로 중앙 서버 스스로 단일 고장점(Single Point of Failure)이 되어 서버가 다운 되면 전체 시스템이 멈추게 되고, 지역적으로 일어나는 서비스에 대해 중앙으로 데이터 트래픽을 발생시킨다. 오버레이 네트워크 기반 자율군집형 미들웨어 플랫폼은 자율군집 메커니즘을 적용하여 구성된 유무선 이기종망 환경 하의 오버레이 네트워크를 통해 관리 서버에 의한 중앙 또는 외부의 제어 없이 노드 간 협업에 의해 다양한 센서 장치(생체신호 측정 장비 포함)와 스마트폰, TV, PC 및 외부 시스템 간에 실시간 스트림 데이터를 송수신할 수 있도록 개발된 미들웨어 플랫폼이다. 생체신호 측정 장비로부터 발생한 여러 생체진단신호를 도처에 존재하는 자율군집형 분산 미들웨어 플랫폼인 SoSpR(Self-organizing Software-platform Router)로 관리 서버의 중재없이 자율적으로 실시간 전달 및 저장하고 동시에 복수개의 다양한 수신 단말에서 가까운 SoSpR로부터 실시간 수신 및 재생 시킬 수 있다.

Key Words : Self-organization, Vital Signal Remote Monitoring, Real-Time Streaming, Message-Oriented Middleware, Mobile Medical Device, Healthcare Software Platform

ABSTRACT

To transmit vital signal stream of mobile patients remotely, it requires mobility of patient and watcher, sensing function of patient's abnormal symptom and self-organizing service binding of related computing resources. In the existing relative researches, the vital signal stream is transmitted as a centralized approach which exposure the single point of failure itself and incur data traffic to central server although it is localized service. Self-organizing middleware platform based on heterogenous overlay network is a middleware platform which can transmit real-time data from sensor device(including vital signal measure devices) to Smartphone, TV, PC and external

※ “이 논문은 2012학년도 경북대학교 학술연구비에 의하여 연구되었음”, “본 연구는 미래창조과학부 및 한국산업기술평가위원회의 산업융합 원천기술개발사업(SW·컴퓨팅)의 일환으로 수행하였음. [10041145, 자율군집을 지원하는 웹밍형 정보기기 내장 소프트웨어 플랫폼 개발”

♦ 주저자 : 경북대학교 자율군집소프트웨어플랫폼연구센터, hykang@ee.knu.ac.kr, 정회원

° 교신저자 : 경북대학교 전자전기컴퓨터학부 실시간시스템 연구실, sjkang@ee.knu.ac.kr, 종신회원

* 경북대학교 전자전기컴퓨터학부 실시간시스템 연구실, snowflower@ee.knu.ac.kr, 정회원, esssusige@ee.knu.ac.kr, 정회원

** 경북대학교 자율군집소프트웨어플랫폼연구센터, jegar08@knu.ac.kr

논문번호 : KICS2013-04-198, 접수일자 : 2013년 4월 30일, 최종논문접수일자 : 2013년 7월 8일

system through overlay network applied self-organizing mechanism. It can transmit and save vital signal stream from sensor device autonomically without arbitration of management server and several receiving devices can simultaneously receive and display through interaction of nodes in real-time.

I. 서 론

자율군집(Self-organization)이란 외부 또는 중앙의 제어 없이 하위 레벨의 시스템 컴포넌트들 사이의 상호작용만으로 시스템 전체의 구조나 기능(패턴)이 생겨나는 것을 말한다^[1]. 자율군집은 시스템 컴포넌트들이 스스로 상황인지하여 자율적으로 바인딩되는 체계이다. 그리고 완전 분산 및 자율 네트워크 하에서 주변 동료 단말간의 자율 또는 협업에 의해 망을 구성하여 서비스를 제공한다.

‘Connected Patient in Home^[3]’은 환자의 집안에서 실시간 환자의 상태를 모니터링하고 의료 장비에서 의사 및 의료 시스템으로 무선 기술을 이용하여 유연하고 유효하게 데이터 전송을 가능하게 하는 서비스 및 인프라이다. 이러한 기술을 통해 약물 복용을 모니터링 할 수 있고, 의료비를 절감할 수 있으며 환자의 의료 접근성 및 의사와의 상호작용을 향상시킬 수 있다. 그러나 의료 서비스(실시간 생체 신호 모니터링 등)를 제공하기 위해서는 복잡한 설정, 환자와 의사의 이동성 보장 불가, 긴급상황 실시간 대응 불가와 같은 문제점을 안고 있는 실정이다.

이러한 문제점을 해결하기 위해 자율군집형 미들웨어 플랫폼을 적용한다. 즉, 집안에서 환자에 의해 사용되는 기기들이 자율군집 네트워크에서 서로 상호작용을 통해 상황인지하여 필요한 서비스를 무설정으로 제공하고, 이동 중에도 노드 간 핸드오버를 통해 유연한 연결을 보장하며, 미리 설정해 놓은 조건(Opportunistic)에 의해 긴급상황 시점에 실시간 생체신호를 전달한다.

본 논문에서는 자율군집 메커니즘 및 분산 컴퓨팅 기술을 이용하여 실시간 생체신호 전달을 위한 자율군집형 미들웨어 플랫폼을 구축하였다. 생체신호 측정 장비로부터 발생한 여러 생체신호(심전도 스트리밍 데이터 등)를 도처에 존재하는 자율군집형 분산 미들웨어 플랫폼인 SoSpR(Self-organizing Software-platform Router)로 실시간 전달 및 저장하고 복수개의 다양한 모니터링 장치에서 가까운 SoSpR로부터 생체신호를 실시간 수신 및 재생시킬 수 있다.

II. 관련 연구

실시간 생체정보의 전달을 위한 미들웨어 플랫폼 개발을 위해 자율군집 및 생체신호 모니터링과 관련된 연구를 살펴본다.

2.1. 자율군집 메커니즘

자율군집 메커니즘은 관련 컴퓨팅 간에 자율적인 서비스 바인딩 등에 필요한 기술을 의미한다. 이러한 자율군집 메커니즘을 활용할 수 있는 분야로 미들웨어, 정보시스템 및 관리, 보안, 로보틱스, 네트워크 관리가 있다. 다양한 자율군집 메커니즘 중 수렵 채집(Foraging)^[2]을 적용하여 서비스 라우팅 및 최적화에 적용하고, 종족 분류(Brood sorting)^[2]를 통해 사설 클라우드의 보안 기능을 구현하고, 조형(Molding)^[2] 및 형태형성(Morphogenesis)^[2] 메타포로 자율 군집적으로 동작 서비스를 바인딩하고자 한다.

자율 군집 메커니즘을 사용함으로써 실내외 모든 영역에 서비스를 제공하는 것이 아니라 수신자가 있는 공간(또는 수신자 소지 수신 단말)에만 서비스를 제공할 수 있고, 중앙 통제 및 제어 개념이 없는 완전 분산과 자율 네트워크 및 주변 동료 단말간의 자율 또는 협업에 의한 망 구성 및 서비스를 제공할 수 있다. 또한 페어링, 동기화 비콘 등 사용자가 해야 하는 초기 설정 작업이 없고 설정을 위한 대기시간도 극히 적으며, 단말 간에 느슨한 연결과 기회 기반 통신을 제공할 수 있다^[1,2].

2.2. 생체신호 모니터링

생체신호 측정 및 모니터링에 대한 연구를 진행한 논문 중 Lin Xu^[4]가 제안한 시스템은 시계 모양의 장치에 ECG와 PPG 센서를 넣고 블루투스 통신으로 휴대폰에 생체신호를 전달한다. 이 시스템은 블루투스 통신이 가능한 인접한 장소에서만 모니터링이 가능하고 여러 모니터링 장비로의 1:N 전달이 불가능하다. Andreas K. Triantafyllidis^[5]이 제안한 모니터링 시스템은 센서 장치와 모바일 장치 간 무선으로 생체신호를 보내고 모바일 장치는 HTTP/REST/SOAP 방식으로 외부 시스템과 통신하

는 구조이다. 이 방식은 단발성의 생체신호를 보내고 이를 소셜 기능 등 다른 외부 시스템과 연계하는 데에는 효과적이지만 실시간 스트림 데이터의 빠른 전송을 보장하지는 않는다. Ashwin K. Whitchurch^[6]가 개발한 무선 원격 Point-of-Care 환자 모니터링 시스템은 환자에게 부착한 8개의 센서 장비로부터 획득한 데이터를 블루투스 통신으로 데이터를 보내고 인터넷을 통해 원격의 의료 시스템으로 실시간 전달한다. 이 시스템은 센서 단말과 수신기 간에 블루투스 클래식 통신만 지원(저전력 문제)하며 중앙 집중식이고 플랫폼 종속적(윈도우즈 운영체제)이다. J. P. Tello^[7]는 ECG 및 체온 데이터를 블루투스를 통해 근처 모니터링 단말로 전송하고 중앙의 서버로는 GPRS 또는 WiFi 전송하여 원격에서 웹 어플리케이션으로 액세스할 수 있는 시스템을 설계 및 개발하였다.

2.3. 기존 연구의 한계

기존의 이동 환자 생체 신호 전달을 위한 중앙 집중화된 방식은 중앙 서버 스스로 단일 고장점 (Single Point of Failure)이 되므로 서버가 다운되면 전체 시스템이 멈추게 된다. 그러나 제안하는 자율군집형 미들웨어 플랫폼은 관리되는 중앙 집중 없이 SoSpR들이 오버레이 네트워크상에서 자율적으로 동작하면서 서비스를 제공한다. SoSpR은 다중 홉으로 데이터 전달 시 발생하는 네트워크 자체의 불안정성(Non-deterministic)을 줄이기 위해 기본적으로 원 홉으로 통신하며, 지역적으로 일어나는 서비스에 대해 중앙으로 데이터 트래픽을 발생시키지 않는다.

따라서 각 SoSpR들이 많은 서비스를 공통으로 가지게 됨으로 오버헤드가 증가하는 단점 보다는 자율 군집 개념을 사용함으로써 얻을 수 있는 이점 (중앙 관리 서버 없음, 지역적 서비스 제공, 무설정, 이동성 보장, 확장성 등)이 있으므로 이동 환자 생체 신호 전달에 더 효과적일 것으로 판단된다. 생체 신호 모니터링 관련 기존의 연구는 센서 단말의 데이터를 단순히 수신 단말로 전달하는 형태로서 환자의 이동성 제공, 상황인지에 의한 긴급상황 실시간 대응과 무설정 서비스 제공, 동시에 복수개의 수신 단말로의 생체 신호 모니터링에 한계가 있다. 본 논문에서는 이동 환자의 생체신호를 전달하기 위해 단말 간 자율 또는 협업에 의해 구성되는 오버레이 네트워크 기반 자율군집형 미들웨어를 개발하여 이러한 문제를 해결하고자 한다.

III. 생체신호 전달 시나리오

자율군집형 미들웨어 플랫폼의 응용 시나리오로 이동환자(가정에서 일상생활 중인 환자)가 갑자기 신체 이상징후를 감지하여 원격에 있는 감시자(의사 등)에게 자신의 생체 진단 신호(ECG)를 실시간 전송하고자 하는 사례를 제시한다.

그림 1은 자율군집 미들웨어 플랫폼을 활용하여 생체신호를 전달하는 서비스 시나리오이다. 그림 1의 구성 요소 중 SoSpR은 미들웨어가 설치된 장치의 이동 가능 유무에 따라 고정형SoSpR과 이동형 SoSpR로 구분된다. 고정형SoSpR은 단위공간에 설치되는 앵커 노드로서 실내 위치인지(대표성)와 무선 노드들의 액세스 포인트 역할을 하며, 스마트폰에 설치되는 이동형SoSpR은 문자 메시지 연동, 긴급 알람, 고정형SoSpR 대리자 등의 서비스를 제공한다. 그림 1의 각 단계별 상세 내용은 다음과 같다.

- ① “생체신호(ECG)의 실시간 전달 및 모니터링” 서비스의 시작은 경북대학교 자율군집소프트웨어플랫폼연구센터 및 실시간시스템연구실에서 공동 개발한 PAAR^[9,10]와치(Personal Activity Assistant and Reminding Watch)에 의해 Trigger된다. 즉, 생체신호 측정 장비(ECG 센서 및 WBAN-Hub)와 PAAR와치를 착용한 독거노인이 갑자기 심장통증을 호소하며 PAAR와치의 긴급 버튼을 누르거나 PAAR와치가 위급 상황을 자동 인지하는 상황을 가정한 것이다. 긴급 버튼이 눌러지면 PAAR와치는 BLE(Bluetooth Low Energy)통신으로 이동형SoSpR(스마트폰)로 긴급상황을 알린다.
- ② 이동형SoSpR은 긴급상황을 미리 정의된 연락처 (의사, 병원관계자, 가족)로 SMS를 전송한다.
- ③ 긴급상황 SMS 메시지가 의사의 스마트폰으로 수신된다.
- ④ 긴급상황 SMS 메시지를 수신한 의사는 즉시 해당 환자의 ECG 생체신호 보기를 원할 시, 근처 고정형SoSpR로 생체신호 스트림 서비스를 요청한다.
- ⑤ 고정형SoSpR은 해당 환자 인근에 있는 고정형 SoSpR로 생체신호 서비스 요청 메시지를 전달한다.
- ⑥ 생체신호 서비스 요청 메시지를 수신한 환자 측의 고정형SoSpR는 WBAN-Hub가 ECG 신호의 측정을 시작하도록 메시지를 보낸다.

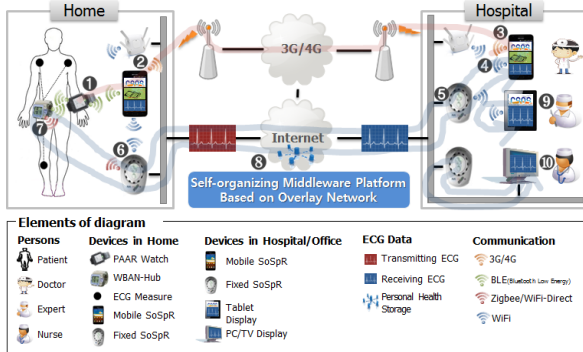


그림 1. 생체신호 전달을 위한 서비스 시나리오
Fig. 1. Service scenario for vital signal transmission

- ⑦ WBAN-Hub는 ECG 생체신호를 측정하면서 데이터를 고정형SoSpR로 계속 보낸다.
- ⑧ 생체신호는 SoSpR들이 자율군집형으로 상호 작용하여 구축된 오버레이 네트워크인 사설 클라우드(Private Federation Cloud)내 분산 개인 헬스 저장소(Distributed Personal Health Record Storage)에 저장되거나 클라이언트로 전달된다.
- ⑨, ⑩ 생체신호가 의사의 스마트폰으로 전달되어 실시간 모니터링 되고 있는 동안에도 병원 관계자, 간호사, 가족도 동시에 생체신호를 태블릿/PC/TV에서 근처 고정형SoSpR을 통해 무선 또는 유선으로 볼 수 있다.

위의 생체신호 전달 서비스 시나리오의 세부 단계 중 ①과 ④에서만 사용자(환자, 의사)의 입력이 요구되고 나머지 모든 단계는 관련 단말에 내장된 자율군집 소프트웨어에서 자동 처리된다.

IV. 요구분석 및 아키텍처

SoSpR은 SoSpR들끼리 자율군집형으로 구성된 오버레이 네트워크를 통해 다양한 센서 장치 (생체신호 측정 장비 포함), 스마트폰, TV, PC 및 외부 시스템 간에 실시간 스트림 데이터를 송수신할 수 있도록 개발된 자율군집형 미들웨어 플랫폼이다. SoSpR은 유무선 이기종망 환경 하에서 관리 서버에 의한 중앙 또는 외부의 제어 없이 노드 간 협업에 의해 실시간 스트리밍 서비스 등 여러 응용 서비스를 제공하는 것을 목표로 하고 있다.

4.1. 요구분석

자율군집형 분산 미들웨어 플랫폼인 SoSpR은 분산 아키텍처로 실시간 스트리밍 데이터를 제어할 수 있어야 하며, 복수개의 여러 수신 장치가 이동

중이라도 모니터링이 가능하고 SoSpR에 의해 생겨난 오버레이 네트워크상에 스트림 데이터를 분산 저장할 수 있어야 한다.

- 스트리밍 제어: 실시간 스트리밍 데이터 프로세싱을 사용자에게 의한 시작, 종료, 중지 및 재개를 할 수 있어야 한다. 또한 런타임 환경에서 여러 센서 장치로부터 소스 스트림의 추가, 변경, 제거를 할 수 있어야 한다.
- Publish & Subscribe 서비스: 하나의 데이터 스트림이 동시에 복수개의 모니터링 장치로 전달되어야 한다.
- 무설정(Zero-Configuration): 사용자가 해야 하는 초기 설정 작업이 없고, 설정을 위한 대기시간도 무시할 수 있어야 한다.
- 이동성 보장 스트리밍: 송신 및 수신 장치 모두 이동 중에도 다른 가까운 SoSpR에 의해 스트림 데이터 전달이 계속 되어야 한다.
- 긴급상황 실시간 대응: 특정 상황을 스스로 인식하고 실시간 알람 메시지를 전달해야 한다.
- 분산 아키텍처: SoSpR은 중앙의 제어가 없는 확장성 있는 자율군집형 분산 아키텍처를 가져야 한다.
- 데이터 저장: SoSpR 내에 스트림 데이터를 저장하고 그대로 재생시킬 수 있어야 한다. (분산 저장된 스트림 데이터는 오버레이 네트워크상에서 분산개인헬스저장소로 확장되어야 한다.)
- 데이터 보안: 물리적 위치 및 사용자 기반 사설 클라우드를 구축해야 한다.
- 시뮬레이션 기능 제공: 네트워크상에 대량의 스트림 데이터를 발생시키는 시뮬레이션 기능을 제공해야 한다.
- 단발성 데이터 처리: 스트림 데이터 외 단발성 데이터 처리 기능을 제공하여야 한다. 그리고 스케줄링된 작업 및 배치 작업 등록 기능도 제공해야 한다.

4.2. 오버레이 네트워크

오버레이 네트워크는 기존의 물리적 또는 논리적으로 존재하는 토폴로지 위에 다시 논리적 링크를 통해 가상의 토폴로지를 재구성하여 특정 기능을 제공하는 네트워크이다. 오버레이 네트워크에서 이웃 노드들은 물리적인 이웃 노드가 아니라 논리적인 이웃노드이다.

그림 2는 SoSpR들이 네트워크 토폴로지를 논리적으로 구성하여 1차 오버레이 네트워크를 생성하

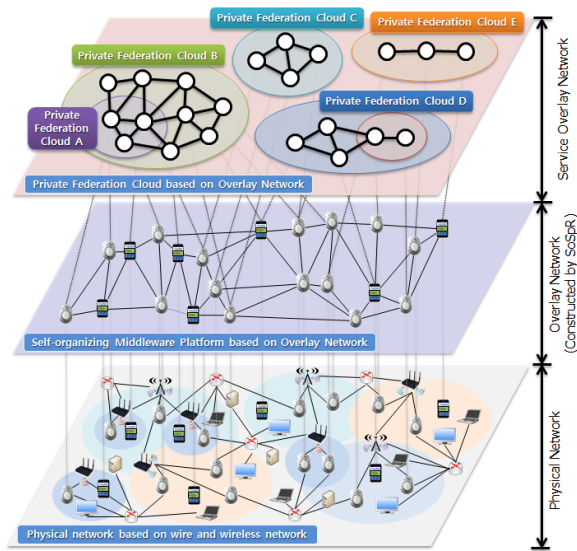


그림 2. 자율군집 미들웨어(SoSpR)로 구성된 오버레이 네트워크
Fig. 2. Overlay Network Constructed by Self-organizing Middleware

고 생체신호 전달과 같은 특정 서비스를 제공하기 위해 특정 위치 및 사용자 정보를 기반으로 2차 오버레이 네트워크를 다시 구성한 모습이다. 1차 오버레이 네트워크는 물리적 네트워크에서 고정형 및 이동형 SoSpR만으로 구성된 네트워크로서, 자율군집형으로 Coordination Service 등을 제공하기 위해서 상호작용하게 된다. 오버레이 네트워크 기반 자율군집 미들웨어 플랫폼은 특정 도메인에서 제공하는 서비스의 기능에 따라 다시 N차 오버레이 네트워크로 유연하게 재구성될 수 있어야 한다. 예를 들면, 그림 2와 같이 생체신호를 전달하기 위해 분산 개인헬스저장소에 저장 및 전달하는 사설 클라우드를 논리적으로 재구성할 수 있도록 지원해야 한다.

4.3. 아키텍처 스택

SoSpR은 위의 기능적 요구사항을 만족시키기 위하여 자율군집형 분산 미들웨어 플랫폼이 가져야 하는 비기능적 요구사항^[2]을 바탕으로 아키텍처를 구성하였다. SoSpR 미들웨어 플랫폼은 그림 3과 같이 실시간 스트림 전송 모듈, 협업(상호작용) 모듈, 사용자 인터페이스 모듈, 개발환경(IDE) 모듈로 나눌 수 있다.

실시간 스트림 전송 모듈은 메시징 미들웨어, 서비스 브로커 & 디스커버리, 서비스 라우팅, 데이터 프로세싱, 시뮬레이션으로 구성된다. 메시징 미들웨어는 경량의 소켓 라이브러리인 ZeroMQ^[11]를 사용

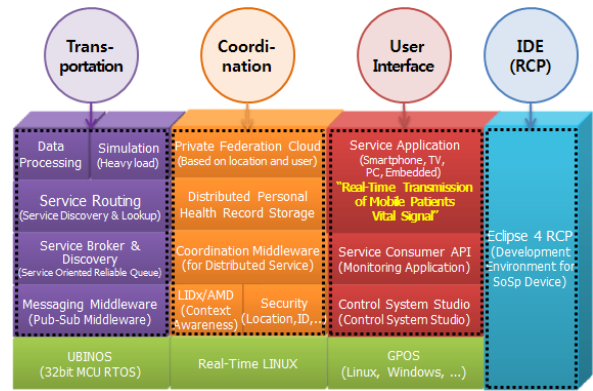


그림 3. SoSpR 아키텍처 스택
Fig. 3. Architecture Stack of SoSpR

하여 Publish & Subscribe 패턴으로 스트림을 송수신하는 역할을 한다. 서비스 브로커 & 디스커버리는 클라이언트가 요구하는 서비스를 생성하거나 이미 생성된 서비스를 검색하는 기능을 수행하며, 서비스 라우팅은 오버레이 네트워크상의 다음 SoSpR로 스트림을 전달하기 위하여 이웃 노드 간 상호작용을 통해 전달 경로를 결정하는 역할을 한다. 데이터 프로세싱에서 실제 스트림 전달이 이루어지며, 시뮬레이션은 SoSpR의 품질을 보장하기 위하여 신뢰성 등 비기능적 요구사항을 가상 시나리오에 기반하여 테스트할 수 있도록 마련된 일종의 테스트 하네스(Test Harness)이다.

협업(상호작용) 모듈은 분산 협업 서비스 (Consensus, Group Management, Leader Election, Presence Protocol)를 제공하기 위한 모듈로서, 분산 서비스에 대한 협업 미들웨어, 분산 개인 헬스 저장소, 사설 클라우드, LIDx(Location ID Exchange)^[8], 위치 및 사용자 ID 기반 보안 컴포넌트로 구성된다. 협업 미들웨어는 Zookeeper^[13]로 구현하였으며 분산 개인 헬스 저장소는 스트림 데이터를 오버레이 네트워크상에 분산 저장하고 사설 클라우드를 통해 서비스를 제공한다. 단말은 LIDx(Location ID Exchange)^[8]로 자동 바인딩되며 위치 및 사용자 ID를 가지고 유효한 사용자 이외의 접근을 제한한다.

사용자 인터페이스 모듈은 현재 CSS/BOY^[15]의 UI 컨트롤을 사용하였다. 이는 Eclipse RCP 기반 IDE에서 플러그인 형태의 SoSpR 라이브러리로 제공한다.

개발환경(IDE) 모듈은 개발자가 SoSpR 어플리케이션을 편리하게 개발할 수 있도록 지원하는 Eclipse RCP 기반 통합개발환경이다. 위의 실시간 스트림 전송 모듈, 협업(상호작용) 모듈, 사용자 인터페이스 모듈의 기능을 API로 제공하며 플러그인

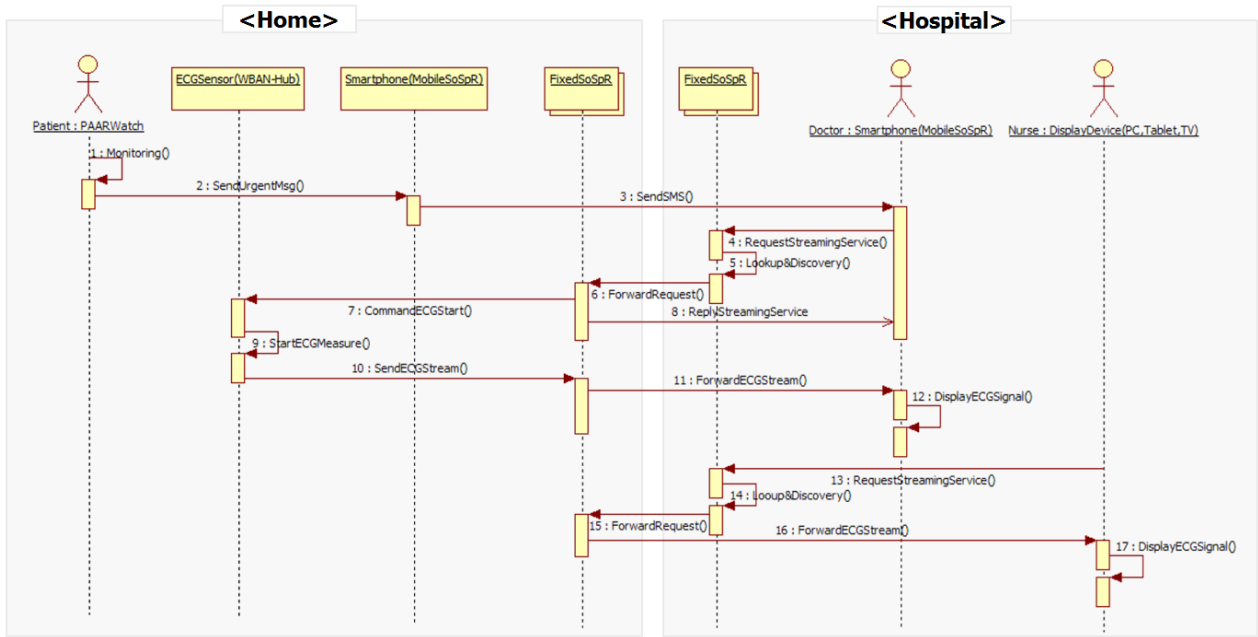


그림 4. ECG 생체신호 전송 서비스의 시퀀스 다이어그램
Fig. 4. Sequence Diagram of ECG Vital Signal Transmission Service

하여 필요한 어플리케이션을 개발 할 수 있도록 제공한다.

V. 설계 및 구현

SoSpR은 그림 3의 아키텍처 스택에 따라 실시간 스트림 전송 모듈, 협업(상호작용) 모듈, 사용자 인터페이스 모듈, 개발환경(IDE) 모듈로 구분하여 설계 및 구현한다.

5.1. 실시간 스트림 전송 모듈

자율군집형 분산 미들웨어 플랫폼은 도처에 존재하는 SoSpR의 미들웨어가 클라이언트가 원하는 요청에 대해 스스로 최적의 서비스를 제공해 주어야 한다. 이러한 자율군집형 분산 아키텍처에서 스트림 전송 서비스의 생성을 위해 SOA(Service Oriented Architecture)구조를 차용하여 “Service Oriented Reliable Queuing^[12]”를 구현하였다. 그림 4는 ECG 생체신호 전송 서비스의 시퀀스 다이어그램이다. PAAR와치는 환자의 응급상황 유무를 모니터링 하고 있다가(1:Monitoring) 응급상황이 발생하면 소 지하고 있는 스마트폰으로 긴급 메시지를 보낸다. (2:SendUrgentMsg) PAAR와치로부터 긴급 메시지를 받으면 미리 설정된 연락처(주치의)의 스마트폰 으로 SMS를 전송하고(3:SendSMS) 메시지를 확인 한 주치의는 근처의 고정형SoSpR로 해당 환자의

ECG 생체신호 스트림 서비스를 요청한다. (4:RequestStreamingService) 고정형SoSpR은 해당 서비스가 이미 존재하는지 살펴보고(5:Lookup &Discovery) 없으면 환자 가까이 있는 고정형 SoSpR로 서비스를 요청 메시지를 포워딩한다. (6:ForwardRequest) 고정형SoSpR이 WBAN-Hub로 ECG 생체신호 측정을 시작하라는 명령을 보내면 (:CommandECGStart) WBAN-Hub는 ECG 신호 측 정을 시작하고(9:StartECGMeasure) 고정형SoSpR로 스트림을 전송한다.(10:SendECGStream) 고정형 SoSpR가 전송받은 스트림을 주치의의 스마트폰으 로 전달하면(11:ForwardECGStream) ECG신호가 화 면에 표시된다.(12:DisplayECGSignal) 이때, 다른 장치 (PC, Tablet, TV 등)도 ECG 신호를 보기 위 해 근처의 고정형SoSpR로 서비스를 요청하면 (13:RequestStreamingService) 해당 서비스를 찾아 서 (14:Lookup & Discovery) 서비스를 가지고 있 는 고정형SoSpR로 포워딩한다.(15:ForwardRequest) 따라서 다른 장치도 해당 고정형SoSpR로부터 ECG 신호를 전달받아(16:ForwardECGStream) 화면에 표 시할 수 있다.(17:DisplayECGSignal)

위의 ECG 생체신호 전송 서비스를 제공하기 위 한 스트림 데이터 전송 패턴은 그림 5와 같다. WBAN-Hub단의 송신 단말은 PUSH-PULL 패턴의 1:1 단방향 통신으로 고정형SoSpR로 데이터를 전 송한다. 스마트폰 등의 수신 단말은 PUB-SUB 패

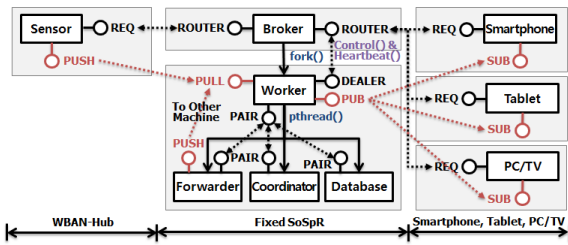


그림 5. 스트림 데이터 전송 패턴
Fig. 5. Transmission Pattern of Stream Data

턴으로 하나의 스트림 데이터를 동시에 여러 단말에서 수신할 수 있는 구조를 가진다. 그리고 고정형 SoSpR은 송수신 단말과 스트림 제어를 위해 REQ-ROUTER 패턴으로 통신하기 때문에 단말이 요청하는 서비스가 다른 고정형SoSpR에 있을 경우 해당 요청이 포워딩될 수 있도록 한다. 또한 고정형 SoSpR은 스트림 전송 처리를 위해 여러 개의 Worker를 생성할 수 있으며 이들 간에 ROUTER-DEALER 패턴으로 IPC 통신하여 제어 명령을 전달하고 Worker는 주기적으로 Heartbeat 메시지를 보내어 Broker가 Worker의 존재 유무를 알 수 있도록 한다. Worker는 스트림 데이터를 다른 고정형SoSpR로 전달하는 쓰레드, 서비스 Lookup & Discovery를 위한 쓰레드, 분산 데이터 베이스에 저장하는 쓰레드를 생성하며 이들 과는 PAIR-PAIR 패턴으로 inproc 통신을 한다.

실시간 스트림 전송 모듈에서 테스트 하네스(Test Harness) 역할을 하는 시뮬레이션을 제외한 모든 기능은 구현 완료된 상태이다.

5.2. 협업(상호작용) 모듈

협업(상호작용) 모듈은 SoSpR 노드들로 오버레이 네트워크 기반 사설 클라우드를 구성하여 데이터 보안을 고려한 편리한 서비스를 제공한다. 분산 어플리케이션을 위한 협업 서비스를 제공하는 Zookeeper^[13]을 기반으로 센서 장치로부터 전달받은 데이터를 분산 저장하는 분산 개인 헬스 저장소 (Distributed Personal Health Record Storage)를 구성하고 연결 단말의 물리적 위치와 사용자의 ID를 기반으로 사설 클라우드(Private Federation Cloud) 형태의 서비스를 제공한다.

분산 개인 헬스 저장소 구축을 위해 데이터베이스 기반 대용량 데이터 처리가 가능한 NoSQL (카산드라^[16], 몽고DB^[17]) 이나 실시간 대용량 데이터의 알고리즘 기반 빠른 처리가 가능한 Storm^[18]등의 활용을 고려하여 구축할 계획이다.

사설 클라우드는 그림 6과 같이 오버레이 네트워크로 구성하여 접속 단말이 원하는 서비스를 제공한다. 데이터 보안을 위해 접속 단말의 물리적 위치 및 사용자 정보를 이용한다. 즉, 병원, 사무실, 집에 여러 대의 SoSpR을 설치하고 각 사용자 또는 그룹 별로 구성된 사설 클라우드 A, B, C에서 원하는 서비스를 안전하게 제공받을 수 있다.

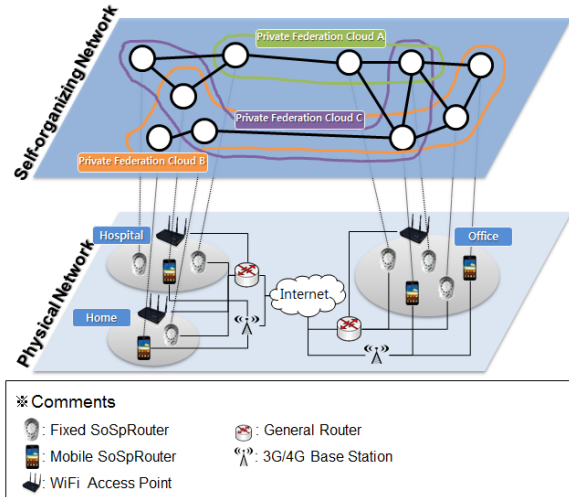


그림 6. 자율군집 네트워크상의 사설 클라우드
Fig. 6. Private Federation Cloud of Self-organizing Network

자율군집 네트워크 하에서 모든 단말은 주변 동료 단말간의 자율 또는 협업에 의해 상황인지하여 자동으로 바인딩하여 필요한 서비스를 무설정으로 제공한다. 사용자가 해야 하는 초기 설정 작업이 없고 설정을 위한 대기시간도 무시할 수 있는 수준을 목표로 한다.

협업(상호작용) 모듈에서 협업 미들웨어, LIDx^[8] (Location ID Exchange), 위치 및 사용자 ID 기반 보안 컴포넌트만 구현 완료된 상태이고, 분산 개인 헬스 저장소와 사설 클라우드 관련 부분은 향후 후속 연구를 통해 구현을 완료할 계획이다.

5.3. 사용자 인터페이스 모듈

사용자 인터페이스 모듈은 수신 단말이 실시간 생체신호 스트림 데이터를 화려하게 표시하도록 해주는 GUI 컨트롤 라이브러리의 모음이다. EPICS^[14], TANGO, TINE 등의 제어 시스템의 모니터링을 위해 사용되는 오픈 소스 소프트웨어 CSS/BOY^[15]를 SoSpR의 실시간 스트림 전송 모듈에서 처리되는 생체신호 전달 데이터 구조에 맞추어 CSS의 IOC (Input/Output Controller) 로직을 수정하였다.

CSS는 CSS Application에서 인지할 수 있도록 PV(Process Variable)라는 고유의 이름을 사용하여 데이터를 송수신한다. SoSpR에서 CSS로의 데이터 전송은 ① Simple DAL(Data Access Layer)로 데이터 전달, ② DAL(Data Access Layer)로 데이터 전달, ③ EPICS에서 DAL로 데이터 전달하는 방법이 있다. ①번과 ②번은 EPICS가 제공하는 DBD(Database Definition)과 DB (Recordset 정의) 데이터 처리 기능을 직접 구현하여야 하지만, 빠른 Throughput, Response Time을 기대할 수 있다. ③번은 EPICS를 통해 CSS와 SoSpR간 의존성을 제거할 수 있고, EPICS API를 이용하여 동적으로 PV 생성 및 삭제가 가능할 뿐만 아니라 여러 CSS Application을 동시 접속할 수 있다. 현재는 ③번의 접근 방법을 통해 구현하였으나 향후 성능 향상을 위해 ①번 또는 ②번의 접근 방법으로 변경할 예정이다.

5.4. SoSpR API

그림 7은 SoSpR의 API를 보여주는 UML 클래스 다이어그램이다. 현재 버전은 SourceStreamsChanger 클래스(생체 신호 측정 장치나 시뮬레이터로부터 SoSpR로 소스 스트림을 부착하거나 분리하는 역할)와 RealtimeStreamSignalsProcessor 클래스 (스트림 신호를 처리하는 태스크를 추가 및 제거하고 데이터 프로세싱을 시작 및 종료, 중지, 재개하고 수신 모니터링 장비의 주소를 추가 및 제거하고 결정된 라우팅 경로에 따라 다음 SoSpR이나 목적지 모니터링 장비로 데이터는 보내는 역할)를 구현한 상태로 이들 2개의 클래스에 대한 API만 제공하고 있다. 자율군집형 분산 서비스 라우팅, 사설 클라우드 등은 일부 구현을 완료하였으며 차기 버전에서 API로 제공할 계획이다.

5.5. 개발환경(IDE) 모듈

개발환경(IDE) 모듈은 개발자가 SoSpR 플랫폼 상에서 SoSpR 서비스 및 클라이언트를 편리하게 개발할 수 있도록 지원하는 Eclipse RCP 기반 통합 개발 환경으로 일반적 요구사항과 특화된 요구사항을 가진다.

일반적 요구사항은 다음과 같다. ① C/C++/Java 코드를 컴파일, 빌드, 디버그, 배포할 수 있어야 한다. ② 내장 소프트웨어를 위한 교차 컴파일 환경을 지원해야 한다. ③ SoSpR 기반 어플리케이션 뿐만 아니라 일반 어플리케이션도 개발할 수 있어야 한

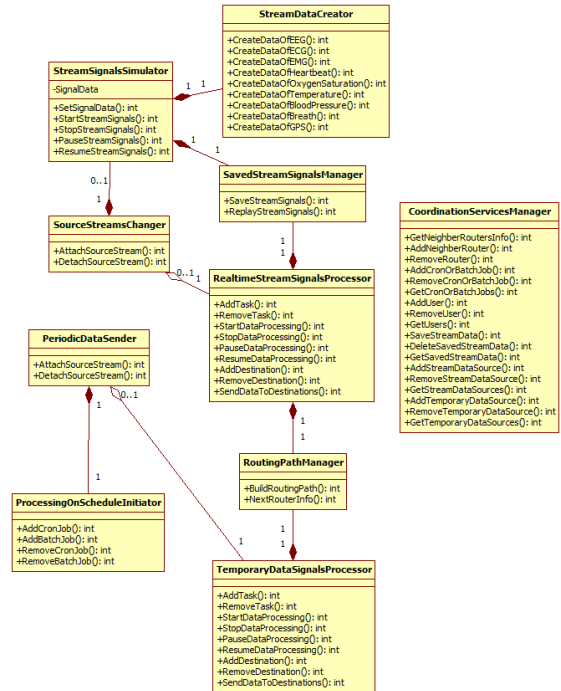


그림 7. SoSpR의 API - UML Class Diagram
Fig. 7. API of SoSpR - UML Class Diagram

다. ④ 리눅스, 윈도우, 안드로이드, 아이폰 등 멀티 플랫폼을 지원해야 한다.

특화된 요구사항으로는 ① SoSp 서비스 및 클라이언트 어플리케이션을 위한 프레임워크 제공 (SoSp API, 템플릿, 문서화, 오픈 커뮤니티 등), ② 타깃 장치에서 표준 빌드를 위한 makefile 제공, ③ 시뮬레이터(SoSp Soft Router) 제공 등이 있다.

그림 8은 이클립스 RCP 기반 통합 개발 환경의 구동 모습을 보여준다. SoSpR 서비스 및 클라이언트를 개발하기 위해 필요한 소프트웨어를 “SoSp” 메뉴에서 자동으로 설치할 수 있으며, SoSpR 기반 서비스(C/C++ Code) 및 클라이언트(Java Code) 프로젝트를 위한 템플릿도 제공한다. 기본적으로 교차 컴파일러, 툴 체인, 원격 디버거, 단위 테스트, 유틸리티 등 일반적인 플러그인들뿐만 아니라 CSS/BOY 기반 GUI 디자이너를 제공하여 디자인 타임에 그래픽, 미터기, 온도계, 탱크 등 여러 가지 위젯을 쉽게 추가할 수 있다.

5.6. SoSpR 구현

현재 버전의 SoSpR은 표 1와 같이 구성되어 있다. 개발 모듈로 SoSpRouter, SoSpWorker, SoSpController, SoSpSerial, SoSpClient, SoSpIDE, SoSpLocator, SoSpSecurity가 있다. 추가로 오픈 소스

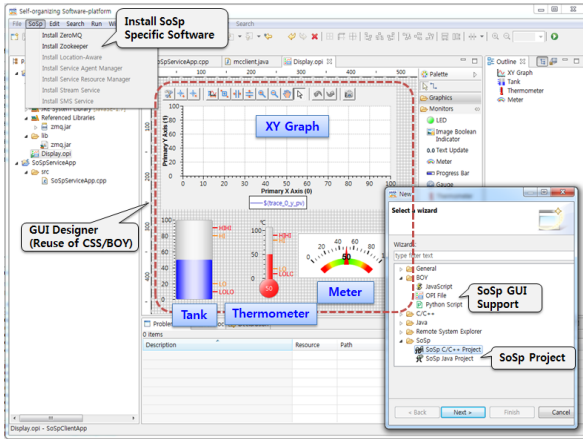


그림 8. 이클립스 RCP 기반 SoSp IDE
Fig. 8. SoSp IDE based on Eclipse RCP

소프트웨어인 ZeroMQ^[11], CZMQ^[11], Zookeeper^[13], CSS^[15], EPICS^[14], Eclipse^[19]를 이용하였다. 현 시점에서 제공 가능한 기능과 제공되지 못한 기능을 구분하면 표 2와 같다.

표 1. SoSpR Ver.0.1.0의 구성
Table 1. Features of SoSpR Ver.0.1.0

Product	Software	Description
Prototype (Ver.0.1.0)	SoSpRouter	- Manage streaming services. - Create streaming service. - Save streaming service. - Forward stream signal. - Remove streaming service.
	SoSpWorker	- Process stream transport service as multi-instance.
	SoSpController	- Control streaming - Manage and monitor coordination service.
	SoSpSerial	- Receive wireless vital signal through Zigbee. - Forward stream signal.
	SoSpClient	- Receive streaming service. - Display as multi-instance.
	SoSpIDE	- Provide IDE for SoSpR Service and Client Project
	SoSpLocator	- Discover and track each other on a local network.
	SoSpSecurity	- Provide security based on location and user identification.
Open Source Software	ZeroMQ 3.2.2	- Lightweight Socket Library.
	CZMQ 1.3.3	- High-level ZeroMQ library.
	Zookeeper 3.4.5	- Provide coordination service for distributed applications
	CSS/BOY 3.1.4	- Using CSS, we can use BOY
	EPICS 3.14.12	- We just only use EPICS base
	Eclipse Juno (4.2)	- Eclipse Rich Client Platform

표 2. SoSpR Ver.0.1.0의 제공 기능 현황
Table 2. Providable Functionality of SoSpR Ver.0.1.0

Module	Sub Module	Providable	Software
Transportation	Data Processing	○	SoSpSerial
	Simulation	×	-
	Service Routing	○	SoSpWorker
	Service Broker & Discovery	○	SoSpRouter
	Messaging Middleware	○	ZeroMQ/CZMQ ^[11]
Coordination	LIDx/AMD ^[8]	○	SoSpLocator
	Security	○	SoSpSecurity
	Private Federation Cloud	×	-
	Distributed Personal Health Record Storage	×	-
	Coordination Middleware	○	Zookeeper ^[13]
User Interface	Service Application	○	SoSpController, SoSpClient
	Service Consumer API	△	SoSpIDE
	Control System Studio	○	EPICS ^[14] , CSS ^[15]
IDE (RCP)	Eclipse RCP Plug-in	○	SoSpIDE
	BOY GUI Toolkit	○	SoSpIDE, CSS ^[15]
	Common C/C++, Java Dev SDK Plug-in	○	Eclipse ^[19]

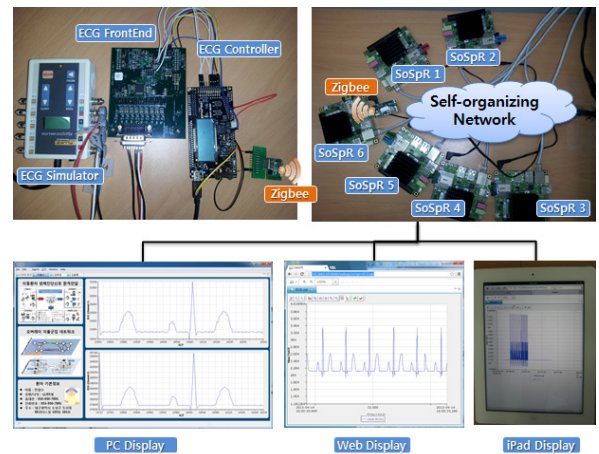


그림 9. SoSpR 구동 환경
Fig. 9. SoSpR Physical Environments

5.7. SoSpR 실행

다음 장의 성능평가를 위해 그림 9과 같이 심전도(ECG)시뮬레이터, 신호전처리기, 신호제어기, 고정형 SoSpR 6대, PC 모니터링 장치 등을 가지고 SoSpR을 구동한다.

심전도(ECG)시뮬레이터는 신호 발생 장치로서 아날로그 ECG 신호를 발생시키고 신호전처리기는 아날로그 신호를 디지털 신호로 변환한다. 신호제어기는 디지털 신호를 실시간으로 SoSpR로 전송하고 SoSpR은 해당 스트림 데이터를 프로세싱하여 모니터링 장치의 요청에 따라 데이터를 전달한다.

VI. 성능 평가

생체 신호는 일반 멀티미디어 스트림과 다르게 각 생체 신호들 사이의 간격이 진단 분석의 기초 자료로 사용되기 때문에 매우 중요하다. 따라서 생체 신호 송신 단말(센서 장치)에서 발생하는 신호 간격은 중간 네트워크상에서 지연없이 동일한 속도를 유지하면서 수신 단말로 전달되어야 한다. 이를 측정하기 위해서는 2msec마다 발생하는 개별적인 신호들 간의 간격을 송신측과 수신측에서 측정하여야 하나 측정의 복잡성으로 인해 대신 간접적으로 송신측의 신호 전송 시간(송신 지속시간)과 수신측의 신호 수신시간(수신 지속시간)을 측정하고 이들 사이의 차이를 데이터 지연시간으로 정의한다. 그리고 실시간 스트림 전송 시 네트워크뿐만 아니라 수신측의 상태에 따라 송신측의 전송 속도를 조정해 줄 필요가 있으므로 이를 위해 송수신 스트림 제어를 위한 별도의 채널을 생성하여 수신 단말의 요청을 처리하게 되는데 이때 송신 단말이 수신 단말의 요청에 의해 스트림을 조정하기까지 걸리는 시간을 데이터 응답시간으로 정의한다.

위 정의에 따라 송신측이 전송하는 생체 신호 간격과 동일한 간격으로 수신 단말이 수신하는지 (데이터 지연시간)를 측정하고, 송신측이 스트림 전송 중에 제어 메시지를 얼마나 빨리 송신측에 전달하는지 (데이터 응답시간)를 측정한다. 다시 말하면, 원 신호와 동일한 신호를 실시간 전달하려면 동일한 속도로 신호를 보내야 한다는 측면에서 데이터 지연시간을, 빠르게 스트림을 제어해야 한다는 측면에서 데이터 반응시간을 측정한다.

$$D(s1,s2,r1,r2)=sub(sub(r2,r1),sub(s2,s1)) \quad (1)$$

$$R(data, req, rep) = sub(rep, req) \quad (2)$$

데이터 지연시간(D)은 센서 장치로부터의 실시간 스트림 데이터 송신 시작시간(s1)과 종료시간(s2) 및 모니터링 장치의 수신 시작시간(r1)과 종료시간(r2)을 측정하여 값을 계산한다(1). 데이터 응답시간(R)은 스트림 데이터 전송(data) 중 SoSpR로 제어 데이터를 송수신 하는데 걸리는 시간(req, rep)을 측정한다(2).

6.1. 성능평가 전제조건

성능평가를 위한 데이터 전송량, 하드웨어 및 소프트웨어 스펙, 전송 프로토콜은 표 3, 4, 5와 같다. 표 3의 Raw Data와 Transmission의 차이는 데이터

전송 포맷에 기인한다. Raw Data 열은 센서장치로부터 SoSpBANHub로 전달받은 데이터로서 ID 8bit, 시퀀스 16bit, 채널당 데이터 24bit로 구성된 고정된 길이의 데이터이고 Transmission 열은 SoSpBANHub에서 SoSpR로 전달하는 데이터로서 문자열로 변경하여 보낼 때의 최대 크기이다.

6.2. 데이터 지연시간(D)

데이터 지연시간의 측정을 위해 약 30초간 ECG 데이터를 전송한다. 송신측의 송신 지속시간과 수신측의 수신 지속시간 측정 결과 그림 10과 같은 그

표 3. ECG 데이터 전송량

Table 3. Transmission Traffic of ECG data

ECG data	Raw Data (bps)	Transmission (bps)
1 ch	24K	64K
2 ch	36K	96K
3 ch	48K	128K
4 ch	60K	160K
5 ch	72K	192K
6 ch	84K	224K
7 ch	96K	256K
8 ch	108K	288K

래프를 획득하였다. 그림 10에서 수신 지속시간을 송신 지속시간으로 빼면 그림 11의 그래프를 얻을 수 있다. 데이터 지연시간은 평균 1.5 msec이다. 측정 결과는 생체 신호 데이터가 1ch ~ 8ch 모두 데이터 지연시간이 5 msec 미만으로 거의 지연없이 수신 단말로 수신됨을 의미하며, 6ch의 지연시간이 상대적으로 높은 이유는 네트워크 단의 불안정성이나 SoSpR들에서 실행되는 운영체제의 스케줄링 등 여러 요인에 의해 발생하는 것으로 보인다. 따라서

표 4. 하드웨어 및 소프트웨어 스펙

Table 4. Hardware and Software Specification

Division	Hardware	Software
SoSpR	- 1.7GHz Quad core ARM Cortex-A9 - 2GB Memory	- Ubuntu/Linaro 12.11 (GNU/Linux 3.0.51, Armv71)
Monitoring	- 3.4GHz i7-3770 - 8GB Memory	- Windows 7 Professional 64bit

표 5. 데이터 전송 프로토콜

Table 5. Data Transmission Protocol

Division	SoSpBANHub → SoSpR	SoSpR → SoSpR	SoSpR → Monitoring
Interface	Wireless(Zigbee)	Wired	Wireless(WiFi)
Protocol	802.15.4	TCP	TCP

수신 단말은 송신측이 전송하는 생체 신호 간격과 거의 동일한 간격으로 생체신호를 수신할 수 있다.

6.3. 데이터 응답시간(R)

스트림 데이터는 전송 중 Flow Control과 같이 수신측의 수신 버퍼환경을 고려하여 적절히 송신측의 데이터양을 조절할 수 있는 채널을 별도로 제공하여야 한다. 그림 12는 ECG 데이터 전송 중 100개의 제어 요청 메시지를 SoSpR을 통해 수신 측에 주고(Request) 받는(Reply) 시간을 측정한 결과를 나타낸 그래프이다. 수신 단말이 스트림 제어를 위해 SoSpR로 100개의 제어 메시지를 보낼 때, 수신 단말의 스트림 제어 요청 메시지 전송 시작 시간과 생체신호 스트림을 전송 중인 송신 단말이 이 메시

지를 모두 수신한 시간과의 차이가 평균 260 msec 이므로 1개의 제어 요청 메시지는 2.6 msec에 송신 단말로 보내진다. 즉, 송신 단말이 수신 단말의 요청에 의해 스트림을 조정하기까지 평균 반응시간이 2.6 msec이다. 따라서 수신단말은 스트림 전송 중에 스트림 속도를 제어할 수 있도록 제어 메시지를 빠르게 보낼 수 있다.

VII. 결 론

SoSpR이 자율군집형으로 구성된 오버레이 네트워크에서 스트림 서비스를 제공하기 위해 노드 간 협업(상호작용)을 통해 해당 서비스를 단말의 요구에 따라 관리 서버의 중재없이 자율적으로 실시간 생체신호를 동시에 여러 단말로 전달할 수 있다. 실시간 스트림 데이터 전송을 위한 분산 메시징 미들웨어, 서비스 브로커 & 디스커버리, 서비스 라우팅 등을 개발하고, 분산 협업 서비스 제공을 위해 협업 미들웨어, 분산 개인 헬스 저장소 등을 구축하여 수신 단말에 ECG 생체신호를 그래프로 표시하였다. 그리고 구현한 오버레이 네트워크 기반 자율군집형 미들웨어 플랫폼의 성능을 평가하고자 데이터 지연 시간(D), 데이터 반응시간(R)을 측정하고 결과를 분석하였다. 이러한 자율군집 메커니즘과 분산 컴퓨팅 기술이 함께 적용된 본 미들웨어 플랫폼은 “생체신호의 실시간 전달 및 모니터링” 뿐만 아니라 다양한 도메인에 활용 및 적용이 가능할 것으로 기대된다.

현재 개발된 버전을 상용화 하기위해 향후, 다음 4가지 기능을 추가로 개발하여 확장하고자 한다.

- 자율군집형 Fault Tolerance: SoSpR 간 상호작용을 통해 장애 대처 능력 확보
- CSS DAL(Data Access Layer) Adaptor: SoSpR과 CSS간 직접 연결하여 스트림 데이터를 송수신하는 Adaptor 개발
- 분산 개인 헬스 저장소: 대용량 데이터 실시간 처리 기술 개발

References

[1] F. Dressler, "A study of self-organization mechanisms in ad hoc and sensor networks," *Comput. Commun.*, vol. 31, no. 13, pp. 3018-3029, Aug. 2008.

[2] F. Dressler, "A study of self-organization mechanisms in ad hoc and sensor networks,"

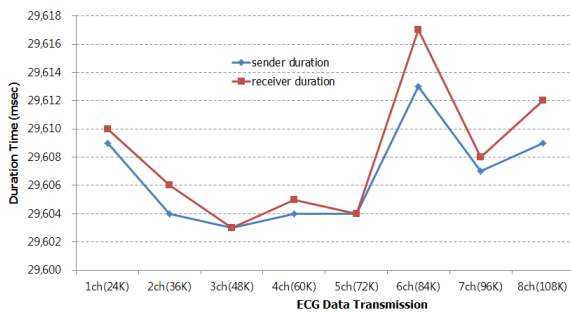


그림 10. ECG 데이터 전송량에 따른 송수신 지속시간
Fig. 10. Transmission Duration Time of Sender and Receiver

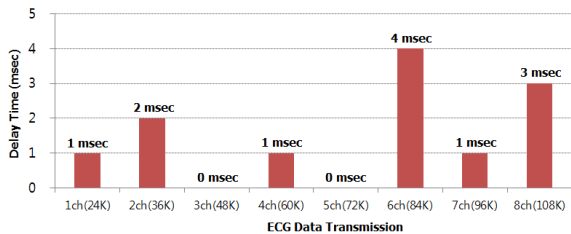


그림 11. ECG 데이터 전송량에 따른 데이터 수신 지연시간
Fig. 11. Delay Time of Receiver according to data traffic

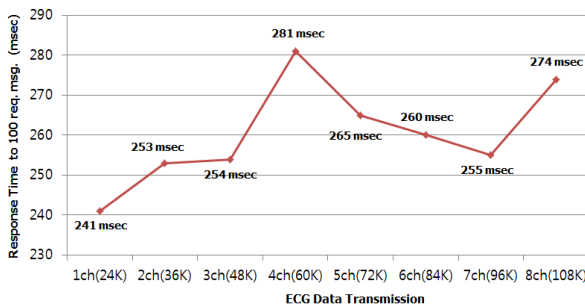


그림 12. ECG 데이터 전송 중 100개 제어 요청 메시지 반응시간
Fig. 12. Response Time to 100 Request Control Message

- Comput. Commun.*, vol. 31, no. 13, pp. 3018-3029, Aug. 2008.
- [3] J. E. Demetriades, R. M. Kolodner, and G. A. Christopherson, *Person-Centered Health Records Toward HealthPeople*, Springer, 2010.
- [4] L. Xu, D. Guo, F. E. H. Tay, and S. Xing, "A wearable vital signs monitoring system for pervasive healthcare," in *Proc. IEEE Conf. Sustainable Utilization Develop. Eng. Tech. (STUDENT)*, pp. 86-89, Petaling Jaya, Malaysia, Nov. 2010.
- [5] A. K. Triantafyllidis, V. G. Koutkias, I. Chouvarda, and N. Maglaveras, "A pervasive health system integrating patient monitoring, status logging, and social sharing," *IEEE J. Biomedical Health Informatics*, vol. 17, no. 1, Jan. 2013.
- [6] A. K. Whitchurch, J. K. Abraham, and V. K. Varadan, "Design and development of a wireless remote point-of-care patient monitoring system," in *Proc. IEEE Region 5 Tech. Conf.*, pp. 163-166, Fayetteville, U.S.A., Apr. 2007.
- [7] J. P. Tello, O. Manjarres, M. Quijano, A. Blanco, F. Varona, and M. Manrique, "Remote monitoring system of ECG and human body temperature signals," *IEEE Latin America Trans.*, vol. 11, no. 1, pp. 314-318, Feb. 2013.
- [8] T. H. Kim, H. G. Jo, J. S. Lee, and S. J. Kang, "A mobile asset tracking system architecture under mobile-stationary co-existing WSNs," *Sensors*, vol. 12, no. 12, pp. 17446-17462, Dec. 2012.
- [9] K. C. Lee and S. J. Kang, "WMDP: Wearable Medical Device Platform supporting opportunistic computing service among co-related device," in *Proc. 8th Int. Symp. Embedded Tech. (ISET2013)*, pp. 111-112, Daegu, Korea, May 2013.
- [10] RTLabs, *Ubinos(2013)*, Retrieved Apr., 6, 2013, from <http://www.ubinos.org>.
- [11] P. Hintjens, *ØMQ - The Guide (2013)*, Retrieved Apr., 6, 2013, from <http://zguide.zeromq.org>.
- [12] P. Hintjens, C. Remes, and G. Goldstein, *7/MDP-Majordomo Protocol*, Retrieved Apr., 6, 2013, from [http://rfc.zeromq.org/spec:7/MDP\(2013\)](http://rfc.zeromq.org/spec:7/MDP(2013)).
- [13] Apache Software Foundation, *ZooKeeper 3.4 Documentation(2013)*, Retrieved Apr., 6, 2013, from <http://zookeeper.apache.org>.
- [14] Argonne National Laboratory, *EPICS Documentation(2013)*, Retrieved Apr., 6, 2013, from <http://www.aps.anl.gov/epics/>.
- [15] DESY, *SNS and BNL CSS Documentation(2013)*, Retrieved Apr., 6, 2013, from <http://css.desy.de/>.
- [16] Apache Software Foundation, *Cassandra Documentation(2013)*, Retrieved Apr., 6, 2013, from <http://cassandra.apache.org/>.
- [17] 10gen, *MongoDB Documentation (2013)*, Retrieved Apr., 6, 2013, from <http://docs.mongodb.org/>.
- [18] Storm, *Storm Documentation(2013)*, Retrieved Apr., 6, 2013, from <http://storm-project.net/>.
- [19] Eclipse, *Eclipse Juno(4.2) Documentation*, Retrieved Jun., 21, 2013, from <http://www.eclipse.org/>.

강 호 영 (Ho-Young Kang)



2007년 2월 아주대학교 정보
및컴퓨터공학부 공학사
2012년 8월 아주대학교 정보
통신공학과 공학석사
2013년 3월~현재 경북대학교
융합소프트웨어학과 박사과
정

2007년~2011년 삼성전자 종합기술원, DMC연구소
연구원

2012년~2013년 대구테크노파크 모바일융합센터 연
구원

2013년~현재 경북대 자율군집소프트웨어연구센터
연구원

<관심분야> 소프트웨어 공학, 소프트웨어 테스트,
애자일 개발방법론

정 설 영 (Seol-Young Jeong)



2001년 2월 동아대학교 컴퓨터 공학과 졸업
2009년 2월 계명대학교 전산교육학과 석사
2009년 3월~현재 경북대학교 전자전기컴퓨터학부 박사과정

2001년~2004년 (주)유진로본 Researcher
2005년~2008년 (주)맥산 Assistant Researcher
<관심분야> 임베디드 시스템, 센서네트워크, RTOS

안 철 수 (Cheol-Soo Ahn)



2013년 2월 경북대학교 전자공학부 공학사
2013년 2월~현재 경북대학교 융합소프트웨어학과 석사과정
<관심분야> 실시간 시스템, 임베디드 시스템, 무선 센서

네트워크

박 유 진 (Yu-Jin Park)



2009년 8월 경북대학교 전자전기컴퓨터학부 공학사
2012년 2월 경북대학교 전자전기컴퓨터학부 공학석사
2012년 2월~현재 경북대학교 전자전기컴퓨터학부 박사과정

<관심분야> 실시간 운영체제, 무선 센서 네트워크, 임베디드 시스템, 실시간 시스템

강 순 주 (Soon-Ju Kang)



1983년 2월 경북대학교 전자공학과 공학사
1985년 2월 한국과학기술원 전자계산학과 공학석사
1995년 2월 한국과학기술원 전자계산학과 공학박사
1985년~1996년 한국원자력연

구소 연구원, 핵인공지능연구실 선임연구원, 전산정보실 실장

2000년~2001년 University of Pennsylvania, Dept. of Computer and Information Science, 객원 연구 교수

1996년~현재 경북대학교 전자전기컴퓨터공학부 정보통신공학전공 정교수

<관심분야> 실시간 시스템, 소프트웨어 공학, 지식 기반 시스템