



농기계의 범용 관리를 위한 K-AgriLink의 구현과 시험

차영욱*, 이종화**, 김춘희***

Implementation and Testing of K-AgriLink for General Management of Agricultural Machinery

Young-Wook Cha*, Jong-Hwa Lee**, and Choon-Hee Kim***

이 논문은 안동대학교 기본 연구지원 사업에 의하여 연구되었음

요 약

본 논문에서는 농기계의 범용 관리를 위한 K-AgriLink의 구조와 구현 및 시험에 대하여 기술한다. K-AgriLink는 통합관리 기능을 하는 스마트매니저, 농기계를 모니터링 하는 부착형의 스마트어댑터 그리고 농기계관리의 하이브리드 앱으로 구성된다. 웹서비스 기반의 농기계관리 인터페이스를 정의하였으며, K-AgriLink를 실험실 모델로 구축하여 농기계관리 기능을 시험하였다. 대단위 농기계들을 관리하는 스마트매니저의 효과적인 구현을 위하여 주요 기능에 대한 성능분석을 수행하였다. 실험실의 PC와 KT의 ucloud 가상서버에 스마트매니저를 구축하여 운영체제 및 컴퓨터 사양에 따른 메시지의 처리시간을 비교하였다. 또한 심각한 농기계의 상태정보에 대하여 스마트매니저에서 안드로이드폰 및 아이폰으로의 알림지연을 측정 및 분석하였다.

Abstract

This paper describes the structure, implementation and testing of K-AgriLink for general management of agricultural machinery. K-AgriLink is comprised of the Smart Manager that performs integrated management functions, the attachable Smart Adapter that monitors agricultural machinery, and a farm machine's management hybrid app. We defined agricultural machine's management interface based on web service, and we built K-AgriLink as a laboratory model to test management functions of agricultural machinery. Performance analysis was performed on main functions for effective implementation of the Smart Manager that manages large farming machines. Smart Managers was established on the laboratory's PC and KT's ucloud virtual server to compare the processing time of messages according to operating systems and computer specifications. In addition, we measured and analyzed the delay of notification from the Smart Manager to Android phones and iPhones for serious agricultural machinery status information.

Keywords

K-AgriLink, agricultural machine, general management, smart manager, smart adapter

* 안동대학교 컴퓨터공학과
- ORCID: <http://orcid.org/0000-0003-1709-7614>
** 금오전자제어
- ORCID: <http://orcid.org/0000-0002-0260-2107>
*** DCU 전자정보통신공학과(교신저자)
- ORCID: <http://orcid.org/0000-0002-7262-0028>

• Received: Sep. 23, 2018, Revised: Nov. 06, 2018, Accepted: Nov. 09, 2018
• Corresponding Author: ChoonHee Kim
Deagu Cyber University, Korea
Tel.: +82-53-850-74, Email: chkim@dcu.ac.kr

1. 서 론

최근 국내의 농업환경은 노동력 부족과 농경지 감소 그리고 자연재해 등으로 많은 어려움을 겪고 있다. 당면한 문제들을 해결하고 지속가능한 농업으로 발전하기 위하여 첨단 기술과의 융합을 통한 농업기술의 혁신이 활발히 진행되고 있다[1]. 글로벌 회사들에서는 ICT 기술을 결합하여 농기계의 원격 모니터링과 진단 서비스를 지원하고 있다. John Deere의 JDLink[2], New Holland의 PLM(Precision Land Management)[3], Yanmar의 SmartAssist[4] 그리고 Kubota의 스마트농업시스템(KSAS, Kubota Smart Agricultural System)[5] 등이 대표적인 농기계관리 시스템이다. 국내의 농기계 회사들은 RS485 인터페이스를 갖는 진단기를 농기계에 연결하여 로컬하게 모니터링 하는 기술은 도입하고 있으나 ICT 기술이 융합된 원격의 농기계관리는 아직 도입하고 있지 않다.

임대사업소나 임대사업자들은 여러 회사의 농기계 제품들을 보유하고 있으며, 이들 농기계에 대하여 특정 회사에 종속되지 않는 범용의 농기계 관리를 요구하고 있다. 본 논문에서는 농기계의 범용 관리를 위하여 ICT와 농기계 기술이 융합된 K-AgriLink의 구조와 구현 및 시험에 대하여 기술한다. K-AgriLink는 통합관리 기능을 하는 스마트매니저, 농기계에 부착되어 시리얼입력으로 상태정보를 모니터링 하는 스마트어댑터 그리고 농기계관리의 하이브리드 앱으로 구성된다. 범용의 농기계관리를 위하여 웹서비스[6]와 CoAP(Constrained Application Protocol)[7]을 이용하는 농기계관리 인터페이스(AMMI, Agricultural Machine's Management Interface)를 정의하였다. 스마트어댑터와 스마트매니저 그리고 하이브리드 앱을 실험실 모델로 구축하였으며, AMMI 인터페이스를 이용하여 K-AgriLink의 농기계 관리 기능을 시험하였다.

대단위 농기계들을 관리하는 스마트매니저의 효과적인 구현을 위하여 주요 기능인 농기계상태 조회와 통보메시지의 처리시간을 측정하였다. 유지보수 및 보안 그리고 비용적인 측면에서 서버 환경을 클라우드 사업자로부터 빌려서 구축하는 추세이다. 실험실의 PC와 KT의 클라우드 서비스인 ucloud[8]

가상서버에 스마트매니저를 구축하여 운영체제와 컴퓨터 사양에 따른 통보메시지의 처리시간을 비교하였다. 또한 스마트매니저가 수신한 긴급한 상태정보에 대하여 푸쉬서버[9]와 스마트기기 별로 알림지연을 측정 및 분석하였다.

II. 관련 연구

2.1 농기계관리의 기술동향

YoonHee Lee의 연구에서는 농기계에 ICT 기술을 적용하여 다양한 정보를 수집하고 활용하는 통합농업정보시스템의 필요성을 강조하며, 클라우드, 로봇 그리고 정밀농업시스템을 스마트농업의 주요 기술 분야로 제시하고 있다. 또한 미래의 농기계 발전 방향으로 자연 환경의 변화에 관계없이 원하는 시기에 농산물을 재배하며, 농업 종사자의 안전과 건강을 보호하는 AI 기술, 농업인의 고령화와 높은 여성 비율에 대응하도록 안전과 조작성의 편의성이 중시되는 기술을 제시하였다[1].

JONH Deere의 JDLink 시스템은 농기계의 위치정보, 운행지역과 운행시간의 제한 서비스 그리고 관제센터로 통보되는 DTCs(Diagnostic Trouble Codes)를 이용한 차량진단 서비스를 제공하고 있다. 이벤트의 우선순위를 색으로 구분하여 스마트폰으로 통보하는 알림 서비스와 정기점검 및 소모품 관리 서비스 그리고 연료 소비량의 분석 서비스를 제공한다[2]. New Holland는 PLM 패키지를 이용하여 농기계를 관리하며, 다양한 센서를 부착하여 생산성 향상을 지원하고 있다. 연료 사용량 및 운행정보가 성능분석 리포트로 변환되어 생산성과 효율성을 개선하는데 이용된다. 웹 페이지를 통하여 농기계의 현재 위치와 동작 상황을 확인할 수 있으며, 설정된 지역을 벗어나거나 설정된 시간 이외에 작동하면 관리자에게 실시간으로 통보한다[3].

Kubota의 KSAS는 농업 경영의 가시화와 농기계에 첨단 ICT 기술을 융합시킨 클라우드 서비스이다. KSAS의 기본코스 서비스는 모든 농사에 활용할 수 있는 농장의 관리, 작업계획과 지시 그리고 작업일지와 같은 농업 경영의 가시화로 효율성을 높이고 생산성 향상을 지원한다. 전문코스는 기본코

스 서비스뿐만 아니라 KSAS 대응 농기계와 연동하여 농기계의 정보관리와 농업관리를 수행한다[4]. Yanmar의 SmartAssist는 GPS안테나와 통신 단말기를 탑재한 농기계에 대하여 작업시간, 주행거리, 연료 소비율, 엔진 회전수, 동력 부하율 등의 정보를 시각화하여 관리할 수 있다. 실시간으로 농기계의 오류 정보를 통지하는 서비스, 기계의 위치를 확인하고 도난을 억제하는 서비스 그리고 3개월마다 기계의 기록을 제출하는 진단보수 서비스를 제공한다[5].

2.2 웹서비스 및 푸쉬서비스 기술동향

SOAP(Simple Object Access Protocol) 기반의 웹서비스는 비즈니스 환경에서 응용 서비스들의 상호 연동을 위하여 시작되었으며, RESTful 웹서비스는 인터넷 서비스 업체들이 응용 개발자들에게 손쉽게 데이터를 제공하기 위한 목적으로 출발하였다. RESTful 웹서비스는 다양한 방식(XML, JSON, HTML, 이미지, ...)으로 표현되는 인터넷의 자원을 HTTP의 기본 요청유형 만으로 접근할 수 있으며, 서버 측에서 클라이언트의 상태를 관리하지 않는 특성을 가진다[6]. Twente대학에서는 CPU와 메모리 사용 그리고 대역폭과 왕복지연 측면의 성능에서 SNMP(Simple Network Management Protocol)에 비하여 웹서비스가 성능에 큰 차이가 없음을 보였다 [10]. IETF에서는 센서노드와 같은 소형 장치의 사물통신(M2M)을 위하여 CoAP를 표준화하였다. CoAP는 HTTP 기능을 재설계한 경량의 웹 전송 프로토콜이다[7].

풀링은 요청 시에 메시지를 전달하며, 푸쉬는 요청이 없어도 메시지를 전달할 수 있는 방식이다. 푸쉬시스템의 구성요소는 푸쉬서버와 스마트기기 그리고 제공자(Provider) 서버이다. 제공자서버는 스마트기기의 레지스터ID와 메시지를 푸쉬서버로 전달하며, 푸쉬서버는 레지스터 ID에 해당하는 스마트기기로 메시지를 푸쉬한다. 푸쉬서버에는 Google의 FCM(Firebase Cloud Messaging)과 GCM, Apple의 APNS(Apple Push Notification Service), NHN의 nPush 그리고 SKT의 AOM(Always On Management) 등이 있다. 푸쉬서버에 따른 푸쉬기술, 메시지 길이와 형

식 및 보관 기간은 표 1과 같다[9].

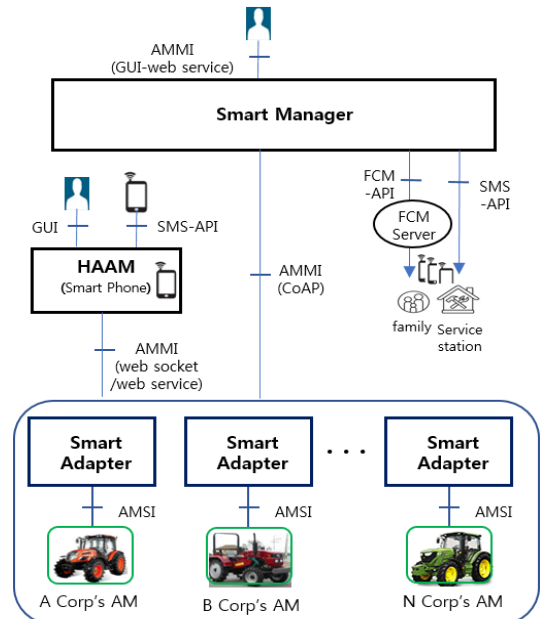
표 1. 푸쉬서버의 비교
Table 1. Comparisons of push servers

	Push technology	Message length	Message format	Storage duration
FCM	TCP/IP	4KB	JSON	4 weeks
GCM	TCP/IP	4KB	JSON	4 weeks
APNS	TCP/IP	256bytes	JSON	unlimited
MPNS	TCP/IP	3KB	XML	30 days
BBPS	WAP	8KB	XML	8 hours

III. 범용 농기계관리를 위한 K-AgriLink의 구성 및 구현

3.1 K-AgriLink의 구성

ICT기술과 융합된 농기계관리는 사전관리를 통한 생산성 향상과 안전성 확보를 제공한다. 농기계관리를 수행하는 K-AgriLink는 그림 1과 같이 스마트매니저, 스마트어댑터 그리고 농기계관리의 하이브리드 앱으로 구성된다.



AMMI: Agricultural Machine's Management Interface
AMSI : Agricultural Machine's Sensing Interface
HAAM: Hybrid App for Agricultural Machine's Management

그림 1. 범용 농기계관리를 위한 K-AgriLink 구성도
Fig. 1. K-AgriLink configuration for general agricultural machinery management

범용의 농기계관리를 위하여 스마트어댑터는 CAN(Controller Area Network)[11] 기술의 탑재 여부와 관계없이 시리얼입력 기반의 AMSI(Agricultural Machine's Sensing Interface)로 상태정보를 모니터링 한다. 통합관리 기능을 하는 스마트매니저는 CoAP 기반의 원격 농기계 인터페이스와 웹서비스 기반의 사용자 인터페이스를 위하여 AMMI를 이용한다. 농기계 정비업체 11곳과 판매업체 12곳의 설문조사를 통하여 우선적으로 요구하는 7개의 농기계관리 항목을 다음과 같이 AMMI에 일차적으로 도입하였다.

- 연료필터의 수분과 배터리 상태
- 엔진의 수온과 부동액 양의 상태
- 엔진오일과 트랜스미션 필터 상태
- 농기계의 안전각도

3.2 스마트어댑터 및 하이브리드 앱

스마트어댑터는 농기계의 상태를 주기적으로 모니터링 하여 안전이나 사전 서비스가 요구되는 경고 사항과 고장 시에 스마트매니저와 운전자의 스마트폰으로 자동 통보한다. 스마트어댑터는 그림 2와 같이 리눅스 환경에서 농기계관리 소프트웨어가 동작하는 메인보드(라즈베리파이3B)[12]와 자체 개발한 인터페이스 보드로 구성된다.

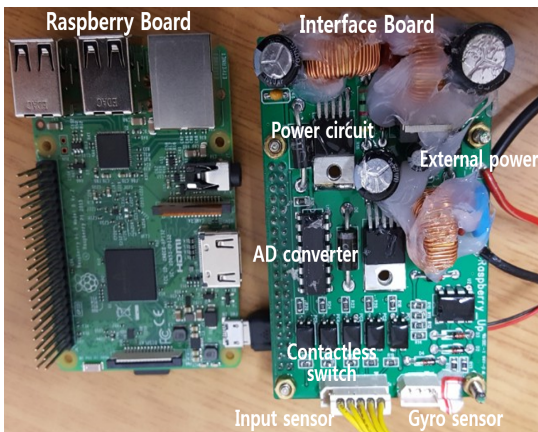


그림 2. 메인보드와 인터페이스 보드로 구현한 스마트어댑터

Fig. 2. Smart adapter's implementation with main and interface boards

인터페이스 보드의 전원 조정부는 12V 또는 24V의 농기계 전원에서 스마트어댑터로 5V의 안정적인 전원을 공급한다. 무접점 스위치로 구현되는 농기계 상태 센싱부는 시리얼입력 기반의 AMSI 인터페이스로 농기계에 설치된 센서들의 상태 정보를 주기적으로 모니터링 한다. 메인보드의 소프트웨어는 운전자 스마트폰의 하이브리드 앱이 요청하는 AMMI의 로그인과 농기계관리를 수행한다. 또한 스마트어댑터에서 진단한 중요 상태정보를 스마트매니저와 스마트폰으로 실시간 통보한다. 통보를 위하여 운전자의 스마트폰과는 웹소켓으로 구현되는 AMMI를 이용하며, 스마트매니저와는 CoAP[7]로 구현되는 AMMI를 이용한다.

운전자의 스마트폰에서 동작하는 농기계관리 하이브리드 앱(HAAM, Hybrid App for Agricultural Machine's Management)는 스마트어댑터의 사용자 인터페이스(GUI)와 상태정보를 처리하는 순수(Native) 앱 기능을 수행한다. 앱 기능으로는 웹소켓(AMMI)로 수신한 농기계의 상태정보를 화면에 출력하며, 심각한 문제가 발생한 경우에 비상 연락 전화번호로 SMS 문자 메시지를 전달한다. 하이브리드 앱의 GUI는 로그인과 농기계관리를 위하여 웹서비스[6]로 구현되는 AMMI를 이용한다. 농기계관리를 위한 하이브리드 앱의 GUI는 다음과 같다.

- 안전모드 설정과 로그인 및 로그아웃, 농기계 상태와 구성정보 관리로 연결되는 메인화면
- 모니터링 체크주기, 알림방법(경광등, 진동), 자이로의 안전각도 기준, 엔진오일 교체 후 운행시간 등을 설정할 수 있는 구성정보 관리화면
- SMS로 농기계 상태의 실시간 알림을 위한 휴대폰 번호의 등록 및 사용자 관리화면
- 농기계 상태 및 이벤트 로그의 조회화면

3.3 스마트매니저

농기계들의 원격 모니터링과 알림 서비스 및 고객과 농기계 구성정보의 관리를 수행하는 스마트매니저의 구현 구조와 인터페이스는 그림 3과 같다. 스마트매니저는 리눅스 Cent OS 7에서 자바(Java) 언어로 구현하였으며, 모듈들 사이는 메서드 호출과 데이터베이스 접근을 위한 내부 인터페이스를 가진다.

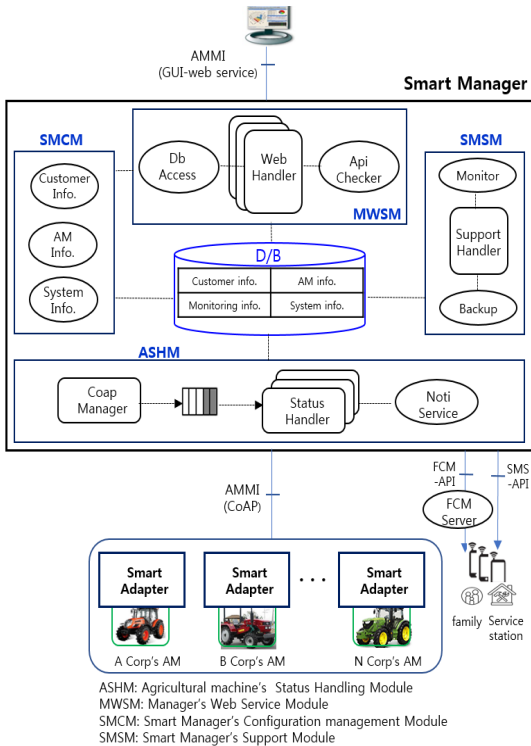


그림 3. 스마트매니저의 구현 구조와 인터페이스
 Fig. 3. Implementation structure and interfaces of smart manager

3.3.1 스마트매니저의 인터페이스

스마트어댑터는 CoAP[7]로 구현되는 AMMI의 통보메시지(notiMsg)를 이용하여 스마트매니저에게 농기계의 상태 정보를 주기적 또는 실시간으로 통보한다. JSON(Java Script Object Notation) 형식으로 정의되는 통보메시지에는 표 2와 같이 차대번호(mano), 비트 단위로 표현되는 농기계의 상태 값(morv) 그리고 엔진오일 교체 후의 운행시간(eohr)이 포함된다.

RESTful 웹서비스 기반의 AMMI를 통하여 고객 및 농기계의 구성정보 관리, 농기계상태의 로그와 시스템 정보의 조회 그리고 알림서비스의 설정을 수행할 수 있다. AMMI는 관리정보의 조회, 생성 그리고 변경과 삭제를 위하여 HTTP메시지의 기본 요청유형을 이용하며, 연구[13]에서 정의하고 있는 URL 사용규칙을 적용하였다. AMMI에 적용되는 요청유형과 URL 및 상태코드는 표 3과 같다[14].

표 2. JSON을 이용한 통보메시지의 정의
 Table 2. Definition of notification message using JSON

AMMI's JSON definition for status notification	<pre> { "notiMsg" :{ "mano": "ABCDEFH12345678", //machine number(max 16) "morv": "0100001", //fuel filter-moisture, //antifreeze amount, battery, //engine water's temperature, //safety state "eohr": "54" //operation time //after engine oil change } } </pre>
--	---

3.3.2 스마트매니저의 구현

스마트매니저의 농기계상태 처리모듈(ASHM: Agricultural machine's Status Handler Module)은 스마트어댑터로부터 수신하는 농기계상태의 통보메시지를 처리한다. ASHM 모듈의 CoopManager와 Status Handler는 Java 쓰레드로 구현된다. CoopManager 쓰레드는 통보메시지의 오류 체크를 수행 후에 Status Handler 쓰레드가 처리하도록 전달하는 분배 기능을 수행한다. 대단위 농기계들로부터 수신하는 통보메시지의 신속한 처리를 위하여 StatusHandler는 다중 쓰레드로 구현되며, CoopManager와의 내부 통신은 메시지 큐를 이용하였다. StatusHandler는 수신한 상태 정보를 로그 형태로 D/B에 저장하며, 긴급한 상태를 알리기 위하여 NotiService 메서드를 호출한다. NotiService는 FCM이나 SMS-API를 이용하여 상태 정보를 가족이나 수리점의 지정된 전화번호로 알린다.

구성관리 모듈(SMCM, Smart Manager's Configuration management Module)은 농기계의 구성정보(차대번호, 소유자 이름 및 휴대폰 번호, 상태)와 고객 정보(이름, 휴대폰 번호, 주소, 소유 중인 농기계 정보)를 관리한다. 또한 스마트매니저의 버전과 IP주소, 운영체제, CPU, 메모리 용량 등의 시스템 정보를 제공한다.

표 3. 사용자 인터페이스(AMMI)의 요청유형과 URL 및 상태코드
Table 3. GUI-AMMI's request types, URLs and status codes

request type	URL(http://Smart Manager's address)	status code	
		normal	error
POST	/login/id&passwd/{id&passwd}: login	201	400
DELETE	/logout/id/{id}: logout	200	400
POST	/customer, [body: customer information]: registration	201	400
DELETE	/customer/tel-number/{tel-number}: deletion of customer information	200	400
GET	/customer/all: query of all customer information /customer/name, tel-number, address or memo/{name, tel-number, address or memo}: query of customer information using search words /customer/page/{page}: query using a page number	200	400
POST	/agri-m, [body: farm machine's information]: registration	201	400
DELETE	/agri-m/serial-number/{serial-number}: deletion of farm machine's information	200	400
GET	/agri-m/all: query of all agricultural machinery information /agri-m/tel-number, serial-number, name or agri-m-type /{tel-number, serial-number, name or agri-m-type}: query of agricultural machine's information using search words	200	400
GET	/agri-m-log/all: query of all farm machines' logs /agri-m-log/start-time&end-time/{start-time&end-time}: query of farm machines' logs using times /agri-m/tel-number, serial-number or name/{tel-number, serial-number or name}: query of farm machines' logs using search words /agri-m-log/page/{page}: query using a page number	200	400
POST	/push/all, [body: message, area, reservation time]: message push to all customers in the selected region at scheduled time /push/select, [body: message, name, phone number]: message push for a specific customer	201	400
PUT	/admin-info, [body: administrator information]: modification	200	400
GET	/system: system information query of smart manager	200	400

스마트매니저의 지원 모듈(SMSM, Smart Manager's Support Module)은 관리정보의 주기적인 백업과 쓰레드의 동작 여부를 모니터링한다. 고객정보와 농기계 구성정보는 주기적 반복 작업을 수행하는 리눅스 데몬(cron)과 “/etc/crontab”을 이용하여 일주일 단위로 백업을 수행한다. 농기계상태의 모니터링 로그는 Maria D/B에서 지원하는 이벤트 스케줄러를 이용하여 일년 단위의 백업을 수행하며, 365일 지난 정보는 자동 삭제되도록 하였다.

Apache Tomcat[12]로 구현되는 웹서비스모듈(MWSM, Manager's Web Service Module)은 RESTful 웹서비스 기반 AMMI를 처리한다. MWSM 모듈은 사용자 인터페이스의 각 요청 별로 생성되는 WebHandler 쓰레드와 각 쓰레드가 호출하는 ApiChecker 및 DbAccess 메서드로 구성된다. ApiChecker는 HTTP 요청메시지에 대하여 Java의 Matches 클래스로 URL과 메시지에 포함되어 있는 JSON 정보의 유효성을 검증한다. 검증이 성공하면

관리정보의 처리를 위하여 요청유형, JSON 정보와 URL에서 추출한 인자로 DbAccess 메서드를 호출하여 Maria D/B에 액세스한다. 그림 4는 차대번호, 이름, 휴대폰 번호, 농기계 종류 그리고 농기계의 전체 상태를 나타내는 GUI 화면이다.

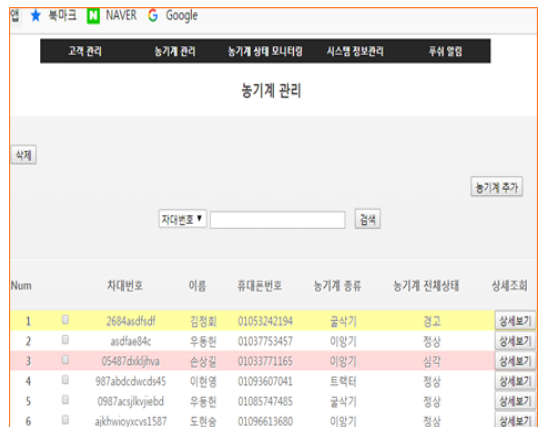


그림 4. 농기계관리의 GUI 화면
Fig. 4. GUI screen for agricultural machine's management

IV. 스마트매니저의 시험

스마트매니저의 구현구조와 운영체제 및 컴퓨터 사양에 따른 메시지 처리시간의 측정을 위하여 그림 5와 같은 시험환경을 구축하였다.

스마트매니저는 KT ucloud의 가상서버와 실험실 PC의 윈도우 및 리눅스 환경에서 동작한다. Apache JMeter를 이용하여 웹서비스 기반의 사용자 인터페이스(AMMI)를 생성하며, 자체 구현한 CoapGen을 이용하여 농기계상태 통보메시지를 생성한다. Apache JMeter는 HTTP, HTTPS, FTP 및 SOAP 등의 다양한 프로토콜과 대단위 요청을 동시에 전송할 수 있으며, 요청 전송시간과 응답 도착시간을 측정할 수 있는 웹 프로그램의 성능 테스트 툴이다[16]. Apache JMeter와 CoAPGen은 5대의 PC에서 동작하며, 각 PC에서 균등하게 시험용 메시지를 생성하도록 하였다. 푸쉬서버와 안드로이드 및 iOS 환경에서 동작하는 스마트기기는 알림 서비스에 대한 시험과 알림 지연을 측정하기 위하여 이용하였다.

4.1 농기계상태 조회 및 통보메시지 처리시간

4.1.1 농기계상태 통보메시지의 처리시간

스마트어댑터가 농기계의 상태를 원격으로 통보하기 위해서는 LPWAN(Low Power Wide Area Network) 통신이 요구되나 실험실 모델의 구현에서는 WiFi 통신을 이용하여 시험하였다. 스마트매니저의 농기계상태 처리모듈(ASHM)은 수많은 농기계

가 통보하는 CoAP 기반의 통보메시지를 실시간으로 처리하여야 한다. 통보메시지의 도착율은 스마트매니저가 관리하는 전체 농기계 수와 스마트어댑터의 통보 주기에 따라 결정되는 변수이다. 10만대의 농기계가 15분에서 90분 사이의 통보 주기를 갖는 것으로 가정하여 메시지의 초당 도착율을 20에서 100까지로 정하였다. CoapGen은 지수분포의 도착율을 갖는 통보메시지를 생성하여 UDP소켓으로 스마트매니저에게 전달한다.

그림 6은 ASHM 모듈의 StatusHandler 쓰레드 수와 메시지의 도착율에 따른 처리시간을 나타낸다. 1회 측정 시에 10,000개의 시험용 메시지를 생성하였으며, 그래프는 3회 측정된 처리시간의 평균으로 나타내었다. 처리시간은 ASHM모듈에 통보메시지가 도착하여 StatusHandler 쓰레드에서 상태를 로그 데이터로 저장하기까지의 시간으로 정의하였다.

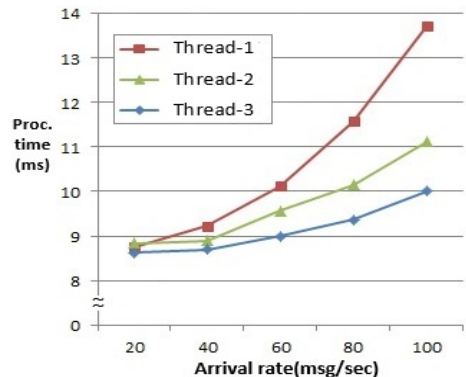


그림 6. 쓰레드 수에 따른 통보메시지 처리시간
Fig. 6. Notification message processing times according to the number of threads

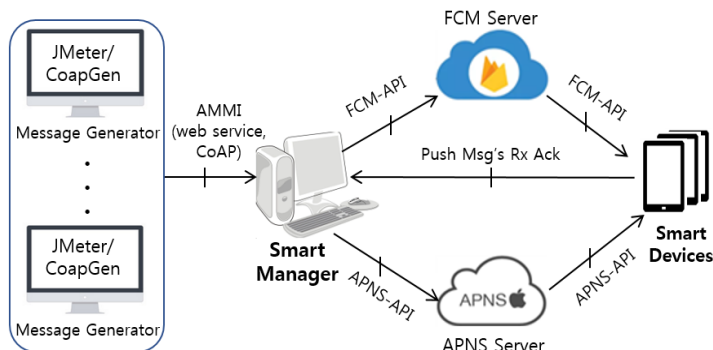


그림 5. 스마트매니저를 위한 시험환경
Fig. 5. Test environment for smart manager

20 이하의 낮은 도착율에서는 쓰레드 개수와 관계없이 처리시간이 비슷하였다. 메시지 도착 시에 쓰레드가 생성되는 동적구조는 처리시간의 장점을 지나 동작의 복잡성과 높은 도착율 시에 과도하게 생성된 쓰레드에 의하여 시스템 부하가 증가할 수 있다. 쓰레드가 3개인 경우에 도착율 100에서의 처리시간은 9.96ms로 측정되었다. 도착율 100에서의 9.96ms는 메시지 간의 도착시간과 유사한 성능이므로 StatusHandler는 3개의 쓰레드로 구현하였다.

4.1.2 농기계상태 조회메시지의 처리시간

농기계관리를 위하여 정의한 웹서비스 기반의 AMMI에 대하여 Apache JMeter[16]를 이용하여 스마트매니저에서의 조회메시지의 처리시간을 측정하였다. 처리시간은 조회 요청메시지의 수신 시간부터 응답메시지의 전송 시간까지로 정의한다. 그림 7은 데이터베이스의 연결방법과 요청메시지의 도착율에 따른 농기계상태 조회의 처리시간을 측정한 결과이다. 조회메시지의 도착율은 상태 통보메시지의 도착율과 동일하게 가정하였다.

DBCP(Database Connection Pool) 방법에 대하여 연결 풀의 수를 10개와 20개인 경우로 구분하여 측정하였으나, 풀의 수에 따른 처리시간은 크게 차이가 없었다. 조회메시지의 처리시간에 있어서 DBCP를 이용하는 방법이 JDBC(Java Database Connectivity)보다 전체적으로 42% 정도 우수하였다. JDBC와 DBCP 방법 모두에서 메시지 도착율의 증가에 따른 처리시간의 차이는 크지 않음을 확인하였다. 이는 웹서비스모듈(MWSM)의 구현에 이용한 Apache Tomcat [15]의 버전이 9.0이며, 기본적으로 설정되어 있는 쓰레드 풀의 최대 수가 200개이므로 100까지의 도착율을 충분히 처리할 수 있었던 것으로 판단된다.

4.2 동작 환경에 따른 통보메시지의 처리시간

그림 8은 실험실의 PC와 KT ucloud의 가상서버에 스마트매니저를 구축하여 운영체제와 컴퓨터 사양에 따른 통보메시지의 처리시간을 나타낸다. 스마트매니저는 DBCP의 풀이 20개이며, StatusHandler 쓰레드가 3개인 환경에서 동작하며, 시험에 사용한

PC의 사양과 운영체제는 다음과 같다.

- KT ucloud 가상서버(Cent OS 7, 4GB 메모리, Intel Xeon E5-2680의 12 코어 중에서 2개의 코어 -2.5GHz 할당)
- 실험실 PC(Cent OS 7, 4GB 메모리, Intel Core Duo E8400, 3GHz 듀얼코어)
- 실험실 PC(윈도우 10, 4GB 메모리, Intel Core Duo E8400, 3GHz 듀얼코어)
- 실험실 PC(윈도우 10, 4GB 메모리, Intel i3-6100, 3.7GHz 듀얼코어)

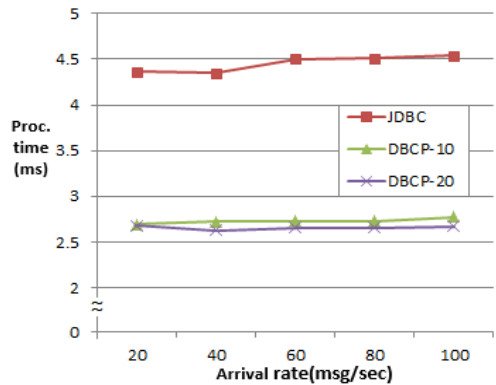


그림 7. 데이터베이스 연결방법에 따른 조회메시지 처리시간

Fig. 7. Query message processing times according to databases connection methods

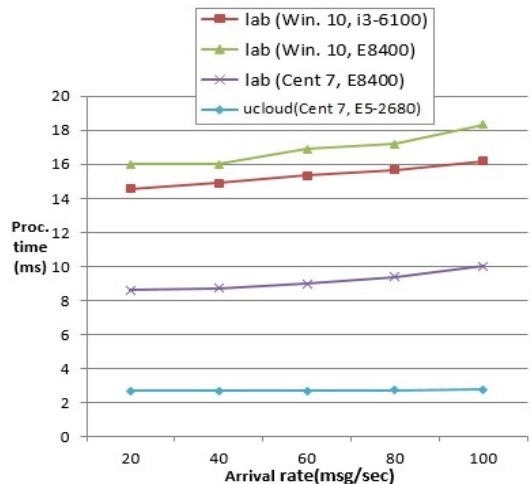


그림 8. 스마트매니저의 동작환경에 따른 통보메시지의 처리시간

Fig. 8. Notification message processing times according to smart manager's operational environment

도착을 100에서 실험실 리눅스 PC(E8400)의 처리 시간은 동일한 H/W 환경의 실험실 윈도우 PC에 비하여 약 1.8배 정도 우수하였으며, 윈도우 PC(i3-6100)의 처리시간에 비하여 약 1.6배 우수하였다. 리눅스 PC에서 동작하는 스마트매니저가 윈도우 PC에서 동작하는 스마트매니저에 비하여 전체적으로 처리시간이 우수함을 알 수 있다. 도착을 100에서 실험실의 윈도우 PC(E8400)과 리눅스 PC(E8400)의 처리시간은 18.3ms와 10.1ms 그리고 KT ucloud의 리눅스 PC에서는 2.8ms로 측정되었다.

Intel Xeon E5-2680의 12 코어 중에서 2개의 코어를 할당하여 생성한 KT ucloud의 CPU는 실험실 PC의 CPU E8400에 비하여 시스템버스 속도와 캐시메모리의 용량 측면에서 월등히 우수한 사양이다. 이러한 이유로 실험실의 리눅스 PC(E8400)과 ucloud 서버가 모두 2개의 코어에서 동작하지만 실험실의 PC보다 ucloud의 서버에서 동작하는 스마트매니저의 처리시간이 3배 이상 우수하였다.

4.3 푸쉬서비스 기반 상태정보의 알림지연

농기계의 심각한 상태정보를 통보메시지로 수신한 경우에 스마트매니저는 푸쉬메시지를 이용하여 가족이나 수리점 기사의 스마트기기로 알림을 수행한다. 실시간 알림의 주요 요구사항인 알림지연을 푸쉬서버 및 스마트기기 별로 총 40회 측정된 평균 지연은 그림 9와 같다.

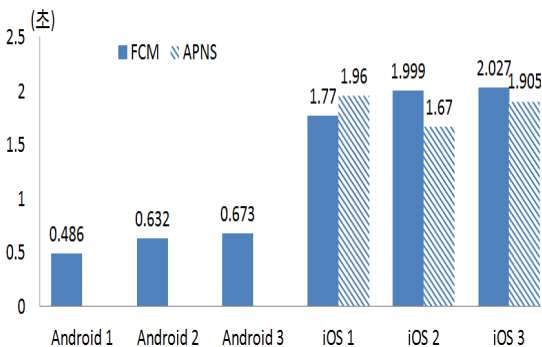


그림 9. 푸쉬서버와 스마트기기에 따른 알림지연

Fig. 9. Notification delays according to push servers and smart devices

알림지연의 측정을 위하여 메시지 생성기 CoapGen은 20초마다 긴급 상황의 통보메시지를 스마트매니저로 전달하며, 상태정보를 수신한 스마트기기는 스마트매니저로 푸쉬메시지의 수신확인을 응답하도록 하였다. 알림지연은 스마트매니저가 푸쉬서버로 푸쉬메시지를 전달한 시간부터 스마트기기가 전달한 수신확인의 도착시간까지로 정의한다. 푸쉬서버는 FCM서버와 APNS서버를 이용하며 스마트기기는 동일한 환경에서 안드로이드폰 3대(Galaxy S6, Galaxy Note4, Galaxy A5)와 아이폰 3대(6S Rose Gold, 6S Space Gray, iPhone 5)를 이용하였다.

스마트매니저에 FCM-API만을 구현하여 안드로이드폰과 아이폰으로 상태정보를 푸쉬하는 경우에 아이폰의 평균 알림지연은 1.932초로 안드로이드폰의 지연(0.597초)에 비하여 1.335초 크게 측정되었다. 아이폰 전용의 APNS-API를 스마트매니저에 추가로 구현하여 측정한 아이폰의 알림지연은 1.845초 이었다. FCM서버를 이용하는 경우와 APNS서버를 이용하는 경우에 대한 아이폰 알림지연의 차이는 0.087초로 크게 차이가 없음을 알 수 있었다. 스마트매니저의 단순화를 위하여 APNS-API를 도입하지 않았으며, 아이폰과 안드로이드폰으로 상태정보를 모두 푸쉬할 수 있는 FCM-API로 알림서비스를 구현하였다.

V. 결 론

본 논문에서는 범용의 농기계 관리를 위하여 스마트매니저, 스마트어댑터 그리고 농기계관리의 하이브리드 앱으로 구성되는 K-AgriLink의 구조와 구현 및 시험에 대하여 기술하였다. 웹서비스와 CoAP를 이용하는 AMMI를 정의하였으며, K-AgriLink를 실험실 모델로 구축하여 농기계관리 기능을 시험하였다.

대단위 농기계들을 관리하는 스마트매니저의 효과적인 구현을 위하여 데이터베이스 연결방법과 쓰레드의 수에 따른 농기계상태 조회와 통보메시지의 처리시간을 측정하였다. 또한 실험실의 PC와 KT의 ucloud 가상서버에 스마트매니저를 구축하여 운영체제 및 컴퓨터 사양에 따른 통보메시지의 처리시간을 비교하였다. 리눅스 PC에서 동작하는 스마트매

니저가 윈도우 PC의 스마트매니저에 비하여 전체적으로 처리시간이 우수하였다. 실험실 PC와 KT ucloud 서버가 모두 2개의 코어에서 동작하지만 ucloud 서버의 H/W는 실험실 PC에 비하여 월등히 우수한 사양이다. 이러한 이유로 실험실의 리눅스 PC보다 ucloud의 서버에서 동작하는 스마트매니저의 처리시간이 3배 이상 우수하였다.

실시간 알림의 주요 요구사항인 알림지연을 푸쉬 서버와 스마트기기 별로 측정하였다. 스마트매니저에서 FCM-API로 푸쉬하는 경우에 안드로이드폰에 비하여 아이폰의 평균 알림지연이 1.335초 크게 측정되었으며, FCM서버와 APNS서버를 이용하는 경우에 대한 아이폰의 알림지연은 0.087초로 크게 차이가 없었다. 스마트매니저의 단순화를 위하여 APNS-API를 도입하지 않았으며, 아이폰과 안드로이드폰으로 농기계의 상태정보를 모두 푸쉬할 수 있는 FCM-API로 알림서비스를 구현하였다.

추후 연구사항으로는 K-AgriLink를 LPWAN 환경에 구축하여 통보주기와 농기계의 수에 따른 스마트매니저의 처리시간을 측정 및 분석할 계획이다.

References

- [1] Yoon-Hee Lee, "Advanced Science of Multi-purpose Farming Machines and ICT-based Agriculture", KISTI MARKET REPORT, Jun. 2016.
- [2] Jone Deere's JDLink, https://www.deere.com/en_INT/docs/html/brochures/publication.html?id=3bb2b86d#1, [accessed June 20. 2018]
- [3] New Holland's PLM, <http://d3u1quraki94yp.cloudfront.net/nhag/apac/en/assets/pdf/precision-land-management/precision-land-management-138009-inb-uk.pdf>, [accessed June 20. 2018]
- [4] Yanmar SmartAssist, https://www.yanmar.com/global/technology/smart_assist.html, [accessed Jul. 18. 2018]
- [5] Kubota KSAS, <https://ksas.kubota.co.jp/>, [accessed July 18. 2018]
- [6] Y. M. Park, A. K. Moon, H. K. Yoo, Y. C. Jung, and S. K. Kim, "SOAP-based Web Services vs. RESTful Web Services", *Electronics and Telecommunications Trends*, Vol. 25, No. 2, pp. 112-120, Apr. 2010.
- [7] Angelo P. Castellani, Mattia Gheda, Nicola Bui, Michele Rossi, and Michele Zorzi, "Web Services for the Internet of Things through CoAP and EXI", *IEEE International Conference on Communications Workshops (ICC)*, pp. 1-6, Jun. 2011.
- [8] KT ucloud biz, <https://ucloudbiz.kt.com/>, [accessed July 22. 2018]
- [9] Na Li, Yanhui Du, and Guangxuan Chen, "Survey of Cloud Messaging Push Notification Service", *International Conference on Information Science and Cloud Computing Companion*, pp. 273-279, Dec. 2013.
- [10] Aiko Pras, Thomas Dreviers, Remco van de Meent, and Dick Quartel, "Comparing the Performance of SNMP and Web Services-Based Management", *IEEE Transactions on Network and Service Management*, Vol. 1, No. 2, pp. 72-82, Dec. 2004.
- [11] Won-Seung Choi and Jin-Gon Shon, "A Study on Reducing CAN Data Traffic for Improving Vehicle Safety", *Korean Institute of Information Technology, The Journal of Korean Institute of Information Technology* Vol. 14, No. 11, pp. 87-92, Nov. 2016.
- [12] Raspberry Pi, <https://www.raspberrypi.org/>, [accessed May 19, 2018]
- [13] 7 Rules for REST API URI Design, <https://blog.restcase.com/7-rules-for-rest-api-uri-design/>, [accessed June 20, 2018]
- [14] S. D. Fan, Y. W. Cha, and C. H. Kim, "Definition of RESTful Web Service for CCTV Sites, and Performance Analysis and Implementation of Integrated Management System", *The Journal of Korean Institute of Information Technology*, Vol. 13, No. 1, pp. 79-86, Jan. 2015.
- [15] Apache Tomcat, <http://tomcat.apache.org>, [accessed

Jun. 25, 2018]

[16] Apache JMeter, Apache Software Foundation,
<http://jmeter.apache.org/>, [accessed Jul. 17. 2018]

저자소개

차 영 옥 (Young-Wook Cha)



1987년 : 경북대학교 전자공학과
(공학사)
1992년 : 충남대학교 전자통계학과
(공학석사)
1998년 : 경북대학교 컴퓨터공학과
(공학박사)
1987년 ~ 1999년 : 한국전자통신

연구원 선임연구원

2003년 ~ 2004년 : 메사추세츠 주립대학 방문학자

1999년 ~ 현재 : 안동대학교 컴퓨터공학과 교수

관심분야 : 망/시스템 제어 및 관리, 네트워크 보안,
개방형통신망

이 종 화 (Jong-Hwa Lee)



1988년 : 금오공과대학교
전자공학과(공학사)
2004년 : 대구사이버대학교
컴퓨터공학과(이학사)
2008년 : 경북대학교
산업공학과(공학석사)
2018년 : 안동대학교 컴퓨터공학과

(공학박사)

관심분야 : 농업 기계관리, IoT 및 센서 네트워크

김 춘 희 (Choon-Hee Kim)



1988년 : 전남대학교 전산통계학과
(학사)
1992년 : 충남대학교 전자계산학과
(이학석사)
2000년 : 경북대학교 컴퓨터공학과
(공학박사)
1988년 ~ 1995년 : 한국전자통신

연구원 연구원

2002년 ~ 현재 : 대구사이버대학교 전자정보통신공학과
교수

관심분야 : 센서 네트워크, 고속통신망, 망 관리 및 제어