



## Vert.x-Based Active Node Management System for IoT Devices

Jung-Uoong Park, Jong-Cheol Park, Hee-Dong Park\*

*Department of Information & Communication, Korea Nazarene University*

### ABSTRACT

As the number of IoT devices increases, the IoT node management server requires more processing power. Plus, the server should manage more connection links and data at the same time to answer to its clients. This causes heavy loads to the server. Therefore, it is required to deploy IoT node management system with high-speed back-end server techniques. One of concurrent server techniques supporting multiple clients is Vert.x which can overcome scalability problem of multi-thread WAS (Web Application Server) such as Tomcat. In this paper, we proposed and implemented an IoT node management algorithm and system model based on Vert.x framework. The proposed system uses Vert.x framework to make it possible to manage and handle IoT devices and their data more quickly when compared with general multi-thread server techniques, because Vert.x is a high-speed back-end server platform. Plus, the proposed system is designed to make it possible for IoT devices to automatically register and communicate with IoT agent server. Our implementation results show that it is possible to manage IoT devices and their data quickly and automatically without human intervention.

© 2016 KKITS All rights reserved

**KEYWORDS :** IoT, Node management, Vert.x, High-speed, Agent server

**ARTICLE INFO:** Received 3 February 2016, Revised 12 February 2016, Accepted 12 February 2016.

### 1. 서론

\*Corresponding author is with the Department of Information & Communication, Korea Nazarene University, 48 Wolbong-ro Seobuk-gu Cheonan, Chungnam, 31172, KOREA.  
E-mail address: [hdpark@kornu.ac.kr](mailto:hdpark@kornu.ac.kr)

최근 센서 및 IoT 기반 서비스에 대한 요구가 폭발적으로 증가함에 따라 다수의 센서 및 IoT 디바이스를 관리하기 위한 노드 관리 시스템의 구축

이 필수적이다. 이를 위해 다중노드의 실시간 진단 및 데이터 통합 관리[1], 센서를 부착한 차량 노드 관리[2], 대규모 IoT 디바이스들 간의 이질성을 효과적으로 해결하기 위한 프레임워크[3], 및 무선 센서 네트워크에서 노드 장애 관리[4] 등에 관한 방안들이 제안되고 있다. 일반적으로 각 센서 및 IoT 노드들은 지그비, 블루투스, z-wave 등의 저전력, 저비용의 통신 기술을 사용하여 주로 저속의 데이터 전송에 사용되어 왔으나[5-10], 최근에는 H/W 및 통신 기술의 발전과 함께 Wifi 통신 및 TCP/IP를 활용한 고속의 데이터 전송 서비스에도 적극 활용되고 있다[11,12]. 이 때 다수의 노드가 실시간으로 대용량의 데이터를 서버와 지속적으로 통신할 경우에는 이를 처리하는 서버에 많은 부하가 발생한다.

<표 1>에서 보는 바와 같이, 여러 클라이언트를 동시에 처리하기 위한 서버 기술은 Tomcat과 같은 멀티 쓰레드 구조 기반의 서버 기술과 Vert.x 등과 같은 싱글 쓰레드 기반의 서버 기술이 있다. Vert.x 기반 서버는 하나의 쓰레드가 Event Loop를 돌며 순차적으로 처리하는 방식으로 10,000~20,000개 정도의 요청을 처리하는 것이 가능하다[13]. 따라서 클라이언트 요청마다 쓰레드를 발생시켜 서버에 부담을 가중시키는 멀티 쓰레드 서버 기술에 비해 높은 성능과 확장성을 가진다.

표 1. 멀티 쓰레드 서버와 싱글 쓰레드 서버 비교  
Table 1. Comparison of multi-thread and single-thread server

	Multi-thread server	Single-thread server
Thread allocation	one thread per client	one thread to all clients
Server resource usage	proportional to the number of clients	not proportional to the number of clients
Concurrency (clients)	50 ~ 500	tens of thousands or more

본 논문에서는 다수의 센서 및 IoT 디바이스들을 관리하고 이들이 전송하는 대용량 데이터를 효율적으로 처리하기 위한 Vert.x 기반의 능동형 IoT 노드 관리 시스템의 구축 방안 및 동작 알고리즘을 제안하고 개발하였다.

## 2. Vert.x 프레임워크

Vert.x는 JVM (Java Virtual Machine) 상에서 동작하며 Vert.x가 동작하게 되면 Verticle instance가 생성되는데, 이 Verticle instance는 하나의 Event Loop 안에서 동작한다[8]. Vert.x에서는 CPU의 개수에 따라 Event Loop를 생성하고 개발자의 설정에 따라 Event Loop의 개수를 변경할 수 있다.

Vert.x는 싱글 쓰레드 상에서 동시 작업을 처리하기 위해 이벤트 기반으로 동작하는데 이는 Reactor 패턴을 기초로 한다[13-16]. Reactor 패턴은 동시에 요청되는 작업에 대한 이벤트를 적절한 핸들러에게 전달해주는 이벤트 처리 패턴이다. 하지만 Reactor 패턴은 멀티 CPU를 가진 시스템에서 CPU를 다 활용하지 못할 뿐만 아니라 병렬처리가 불가능한 큰 단점이 있다[14].

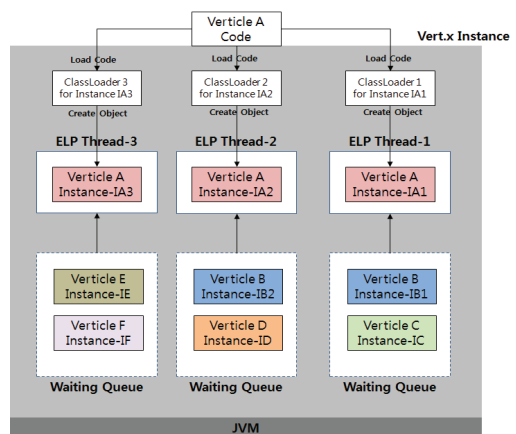


그림 1. 다중 Event Loop로 동작하는 Multi-Reactor 구조[8]  
Figure 1. Multi-Reactor structure with multiple Event loops[8]

Vert.x에서는 이러한 단점을 보완하기 위해 Worker verticle과 Multi-Reactor 구조를 지원한다. 먼저 Multi-Reactor는 병렬처리를 위한 것으로서 <그림 1>에서와 같이 하나 이상의 Event Loop에서 요청을 처리할 수 있다. 이 때 각각의 Event Loop에서 동작하는 Verticle은 각기 다른 인스턴스이다. 이벤트를 순차적으로 처리할 경우 처리가 오래 걸리는 작업이 있으면 병목현상이 발생하는데, 이런 작업들은 Worker verticle에 등록해두고 이 이벤트가 발생하면 별도의 쓰레드에서 이벤트를 처리한 후 이벤트 버스를 통해 결과값을 반환한다[8].

### 3. Vert.x 기반의 능동형 IoT 노드 관리

#### 3.1 제안 시스템 구성

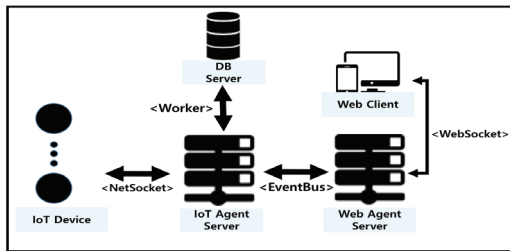


그림 2. 제안 시스템의 구성  
Figure 2. Structure of the proposed system

<그림 2>는 본 논문에서 제안한 Vert.x 기반의 IoT 노드 관리 시스템의 구성도이다. 제안 시스템은 IoT 디바이스, IoT Agent Server, 및 Web Agent Server로 구성되는데, 모두 Vert.x 프레임워크를 기반으로 동작한다. IoT 디바이스는 NetSocket 통신을 통해 IoT Agent Server에 자신을 등록하고 데이터를 주고받으며 Web Agent Server와는 EventBus를 통해 통신한다. 웹 클라이언트는 웹 소켓을 통해 데이터를 받아 웹상에 그 결과를 출력한다.

#### 3.2 시스템 동작 방안

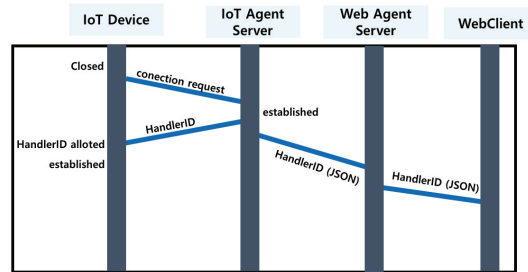


그림 3. 노드 등록 절차  
Figure 3. Node registration procedure

본 논문에서 제안한 IoT 노드 관리 시스템에서는 새로운 IoT 노드가 추가되었을 때 해당 노드의 관리 서버 등록 및 IoT 노드 간 연결이 사람의 개

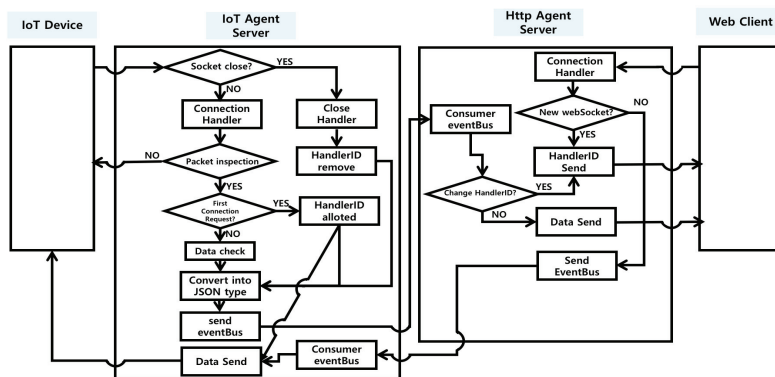


그림 4. 제안 시스템의 동작 알고리즘 및 데이터 전달 과정  
Figure 4. Operating algorithm and data transfer of the proposed system

입 없이 자율적으로 이루어짐으로써 능동적인 노드 관리 및 데이터 전송이 가능하다. 또한 일반 사용자들은 HTML5 및 WebSocket 표준을 통해 실시간으로 IoT 노드를 모니터링 또는 제어할 수 있다.

<그림 3>은 본 논문에서 제안한 IoT 노드 등록 절차를 나타내고 있다. IoT Agent Server는 각각의 IoT Device를 식별하기 위해 각 Device별 고유 HandlerID를 부여한다. 이 후 Web Agent Server에 HandlerID값을 전달하여 Web Client가 Device 연결 여부를 확인할 수 있도록 한다. 이 때 데이터는 JSON형식으로 전달하여 웹상에서 처리한다.

<그림 4>는 제안 시스템의 동작 알고리즘 및 데이터 전달 과정을 구체적으로 나타내고 있다. IoT Agent Server와 Web Agent Server 간 통신은 EventBus를 통해 이루어지며, Web Agent Server와 클라이언트 간에는 소켓 통신을 통해 데이터를 주고받는다. Connection Handler에 소켓연결 이벤트가 발생하면 자동으로 해당 소켓 연결에 대해 HandlerID가 할당되며, 그 후 IoT Device는 이 HandlerID를 사용하여 통신한다.

<그림 5>은 IoT Device의 구체적인 동작 알고리즘을 나타낸다. IoT Device는 사용자의 버튼동작을 감지하면 IoT Agent Server와 자동으로 소켓 연결을 시도하여 <그림 4>에서와 같이 HandlerID를 할당 받은 후 연결된 Socket을 통해 통신하게 된다. 이때 최초 Socket 연결 시도 시 해당 IoT Device의 종류와 Mac 주소 정보를 전송한다. 이후 통신은 할당 받은 HandlerID를 사용하여 SocketHandler에 의해 이루어진다.

<그림 6>는 Web Client의 동작 알고리즘을 나타낸 것으로 이 알고리즘에서 Web Agent와 Web Client는 HTML5 표준인 WebSocket을 이용하여 실시간으로 데이터를 송수신한다. 또한 사용자 뷰 화면 제어를 위해 JQuery 라이브러리를 활용하여 비동기적으로 처리하였다. Web Client에서는 Web

Agent로부터 현재 접속된 IoT Device의 HandlerID와 해당 Device의 종류별 컨트롤 화면을 Jquery의 append 함수를 통해 사용자화면에 추가한다. 또한 Web Agent 서버로부터 소켓 데이터를 수신하면 WebSocket의 onmessage 이벤트가 발생하고 Web Client에서는 노드 갱신 또는 제어 데이터의 갱신인지 판별 후 사용자 뷰를 제어한다. 사용자가 IoT Device에 데이터를 전달 시 수신 IoT Device의 HandlerID와 데이터를 묶어 WebSocket의 send 함수를 통해 전송한다.

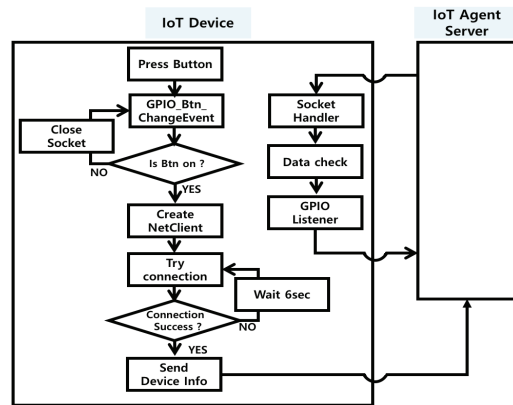


그림 5. IoT Device의 동작 알고리즘  
Figure 5. Operating algorithm of IoT Device

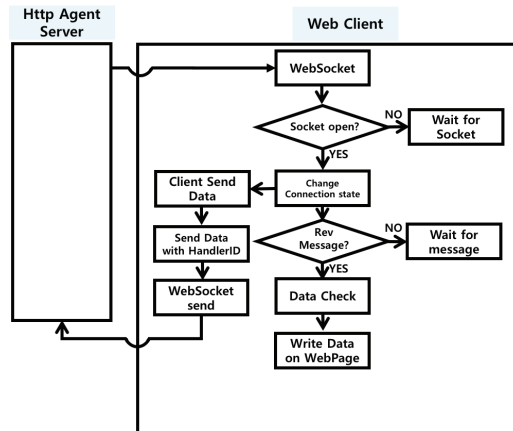


그림 6. Web Client 동작 알고리즘  
Figure 6. Operating algorithm of Web client

#### 4. 구현 내용 및 결과

3장에서 제안한 Vert.x 기반 능동형 IoT 노드 관리 시스템의 서버 및 IoT Device는 각각 애플사의 노트북 및 라즈베리파이를 사용하여 Java/JSP로 구현하였으며, 구체적인 개발 환경은 <표 2>에서와 같다.

표 2. 개발 환경  
Table 2. Development environment

	H/W	OS	JDK/ Vert.x
IoT Server (Agent/Web)	Mac Book Air	MacOS X 10.11	JDK 1.8
IoT Device	RASPBERRY PI 1 MODEL B+	RASPBIAN WHEEZY	Vert.x 3.1.0

<그림 7>은 본 논문에서 구현한 IoT 디바이스와 IoT 노드 관리 시스템의 동작 화면을 보여 주고 있다. IoT Device들은 본 논문에서 제안한 Vert.x 기반 능동형 노드 관리 시스템을 통하여 웹 상에서 신속하고 효율적으로 모니터링 및 제어될 수 있음을 확인하였다. 라즈베리파이에 사용된 센서로는 온습도 센서, 화재감지 센서를 사용하여 2개의 IoT Device를 제작하고 JSP와 Bootstrap을 이용하여 웹 페이지를 구성하였다. 웹에서는 WebSocket을 통해 접속된 IoT Device의 종류와 IoT Device에게 메시지를 전송하는 기능을 구현하였다. 다수의 IoT Device 환경에 대한 서버 성능 테스트를 위하여 서버에 가상의 부하를 적용하여 일반적인 Vert.x 서버의 높은 성능을 기대할 수 있음을 알 수 있었다.

<그림 8>은 구현된 능동형 IoT 노드 관리 시스템 및 동작 실험 결과를 보여 주고 있다. 동작 실험 결과 IoT Device들의 추가 및 삭제가 자동으로 이루어지고, 화재감지 및 온습도 정보 등이 실시간으로 전송됨을 확인할 수 있었다. 사용자가 IoT Device의 버튼을 눌렀을 때 IoT Device는 자동으로

IoT Agent Server와 연결을 수립한 후 HandlerID를 할당 받게 되며 Web Client에서는 접속된 IoT Device를 즉각 접속 디바이스 리스트에 출력한다. 또한 온습도 IoT Device에서 측정된 값은 실시간으로 WebSocket을 통해 웹상에 표시되고, 화재감지 IoT Device는 화재 상황을 가정하여 해당 센서에 열을 가한 결과 그 즉시 Web Client의 Alert 창을 통해 경고 메시지가 출력되는 것을 확인하였다.

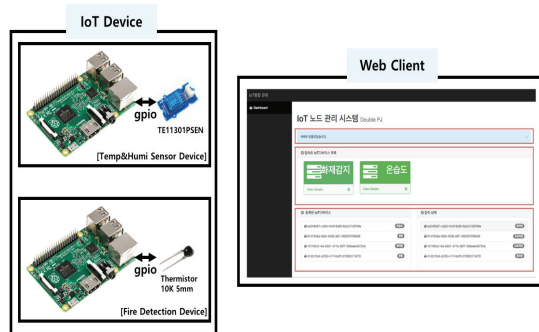


그림 7. IoT Device 및 웹 페이지 구현  
Figure 7. Implementation of IoT Devices and web pages

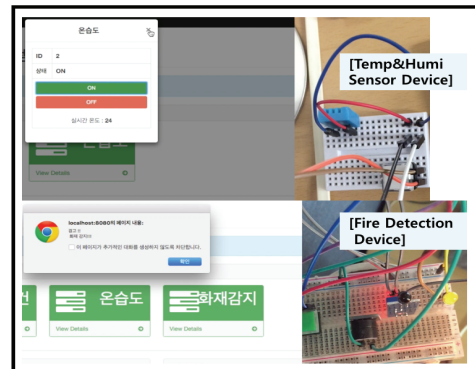


그림 8. 동작 시나리오 및 실험 결과  
Figure 8. Operating scenario and test results

#### 5. 결론

본 논문은 일반적인 멀티쓰레드 기반의 서버에

비해 다수의 클라이언트를 고속으로 처리할 수 있는 Vert.x 프레임워크 기반의 능동형 IoT 디바이스 관리 서버 및 시스템 개발 방안을 제시하여, 다수의 IoT 디바이스 관리 및 고속 데이터 처리가 가능하도록 하였다. 또한 본 논문에서 제안된 시스템은 Java, Ruby, Python, Groovy, Ceylon 등 다양한 언어를 지원하며 범용성 및 확장성이 뛰어나 대규모 클러스터링이나 임베디드 형태로도 동작이 가능하다. 따라서 제안 시스템은 향후 IoT 디바이스 개수의 폭발적인 증가와 함께 고속의 데이터 처리 및 자율 노드 관리를 필요로 하는 대규모 IoT 네트워크에서 그 활용성이 높을 것으로 기대된다.

## References

- [1] M. Kang, *Design of multi-node real-time diagnostic and management system using zigbee sensor network*, Journal of the Institute of Electronics and Information Engineers, Vol. 51, No. 6, pp. 152-161, 2014.
- [2] H.-J. Kim, and H.-H. Yang, *Implementation of mobile node monitoring system for campus vehicle management*, Journal of The Korea Institute of Information and Communication Engineering, Vol. 14, No. 4, pp. 999-1004, 2010.
- [3] D.-Y. Kim, H.-J. La, and S.-D. Kim, *A framework for effectively managing heterogeneity of IoT Devices*, Journal of KISS, Vol. 41, No. 5, pp. 353-366, 2014.
- [4] Y.-W. Koo, S.-Rak Kim, D.-S. Kim, K.-H. Yun, and M.-H. Song, *An node error management technique for efficient service over wireless sensor network*, Proceeding of Korean Society for Internet Information Conference, pp. 121-125, 2009.
- [5] S.-H. Wang, *Techniques and applications of zigbee*, The Journal of The Korean Institute of Communication Sciences, Vol. 30, No. 6, pp. 47-53, 2013.
- [6] H.-D. Park, D.-H. Kim, and H. Park, *An adaptive GTS allocation scheme to increase bandwidth utilization in IEEE 802.15.4*, Journal of Korea Multimedia Society, Vol. 14, No. 2, pp. 219-227, 2011.
- [7] M.-W. Lee, *Study to multi control device of zigbee IBS*, Journal of Korea Multimedia Society, Vol. 12, No. 1, pp. 490-492, 2009.
- [8] IEEE 802.15 Working Group for WPAN: <http://www.ieee802.org/15/>, Dec., 2015.
- [9] IEEE 802.15 WPAN Task Group 4, <http://www.ieee802.org/15/pub/TG4.html>, Jan., 2016.
- [10] D. P. Cerio and J. L. Valenzuela, *Provisioning vehicular services and communications based on a bluetooth sensor network deployment*, Sensors, Vol. 15, No. 6, pp. 12765-12781, 2015.
- [11] IETF 6lowpan working group, <https://datatracker.ietf.org/wg/6lowpan/charter/>, Dec., 2015.
- [12] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, *Transmission of IPv6 Packets over IEEE 802.15.4 networks*, RFC 4944, 2007.
- [13] Introduce Vert.x Framework, <http://bcho.tistory.com/860>, Nov., 2015.
- [14] D. Synodinos, *vert.x - JVM polyglot alternative to node.js*, <http://www.infoq.com/news/2012/05/vertx>, May, 2012.
- [15] D. C. Schmidt, *An object behavioral pattern for demultiplexing and dispatching handles for synchronous events*, Department of Computer Science Washington University, St. Louis, MO, Vol. 12, No. 1, pp. 490-492, 2009.

[16] Vert.x Documentation, <http://vertx.io/docs>, Nov., 2015.

---

## IoT 디바이스를 위한 Vert.x 기반의 능동형 노드 관리 시스템

박정용, 박종철, 박희동

나사렛대학교 정보통신학과

---

### 요 약

IoT 디바이스의 개수가 증가할수록 IoT 노드 관리 서버는 더 높은 처리 능력을 가져야 한다. 또한 동시에 다수의 클라이언트와 연결을 유지하고 데이터를 관리해야 하는데, 이는 서버에 큰 부하로 작용한다. 따라서 고속의 백-엔드 서버 기술을 가진 IoT 노드 관리 시스템이 요구되어 진다. Vert.x는 다수의 클라이언트를 동시에 처리하기 위한 서버 기술 중 Tomcat 과 같은 멀티 쓰레드 구조 기반 WAS (Web Application Server)의 확장성 문제를 극복하고 서버 성능을 향상시키기 위한 서버 기술 중 하나이다. 본 논문에서는 Vert.x 플랫폼 기반의 IoT 노드 관리 알고리즘 및 시스템 모델을 제안하고 구현하였다. 제안된 노드 관리 시스템은 고속의 백-엔드 서버 플랫폼인 Vert.x를 사용함으로써 일반적인 멀티쓰레드 기반의 서버 기술을 사용할 때보다 신속한 IoT 디바이스 및 데이터 관리가 가능하다. 또한 제안된 노드 관리 시스템은 IoT 디바이스들이 자동으로 에이전트 서버(Agent server)에 등록되고 통신을 할 수 있도록 설계되었다. 구현 결과, 제안된 IoT 노드 관리 시스템을 통하여 사람의 개입 없이 자율적이고 신속한 IoT 디바이스 관리가 가능함을 확인하였다.

---

### 감사의 글

본 논문은 2011년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2011-0010627).



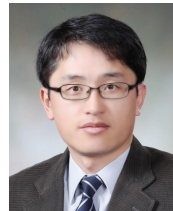
**Jung-Woong Park** has been an undergraduate in the Dept. of Information & Communication at Korea Nazarene University since 2010. His research interests include IoT, smart home, and web contents.

*E-mail address:* myuiong@gmail.com



**Jong-Cheol Park** has been an undergraduate in the Dept. of Information & Communication at Korea Nazarene University since 2011. His research interests include IoT, mobile contents.

*E-mail address:* appleqwr123@gmail.com



**Hee-Dong Park** received the bachelor's degree, M.S. degree, and Ph.D. degree in the Dept. of Electronics from Kyungpook National University in 1993, 1998, and 2005, respectively. He was a professor in the Dept. of Computer Engineering at Pohang College from 1998 to 2007. He has been a professor in the Dept. of Information & Communication at Korea Nazarene University since 2007. His current research interests include IoT, WPAN/WBAN, u-Health, smart home, mobile computing, etc.

*E-mail address:* hdpark@kornu.ac.kr