한국시뮬레이션학회 논문지
Vol. 25, No. 3, pp. 1-14 (2016. 9)
http://dx.doi.org/10.9709/JKSS.2016.25.3.001
ISSN 1225-5904

사물 서브 망과 모바일 인터넷을 연계하는 무선 게이트웨이 타입들의 상태천이모델 기반 설계와 성능 평가

성철제 · 김창화†

State Transition Model-based Design of Wireless Gateway Types to Connect between a Sub-network of Things and Mobile Internet and their Performance Evaluations

Cheol-Je Seong · Changhwa Kim†

ABSTRACT

This paper proposes four general wireless gateway types, which are distinguished by their own processing ways to connect between a wireless sub-network of things and the mobile internet that links mobile network to internet step by step. In this paper, we also design general processing procedures of these four types using the state transition model. Gateways of each types were developed on the basis of the resulted state transition models and their performances were evaluated through several tests, analyzed, and compared each other. As the results of our evaluation, compared with the other types, the type, which combines both of a low-power Sleep-interrupt way and polling ways for receiving data or responses in all the waiting states of a gateway, shows the best performance in all of data transmission real-timeliness, data loss and energy consumption.

Key words : Internet of Things, Sensor Network, Gateway

요 약

본 논문은 사물 무선 서브 망을 모바일 망과 인터넷을 차례로 연계하는 무선 게이트웨이 처리 방식을 중심으로 데이터 유실이 있는 전송 방법과 데이터 무손실 전송 방법으로 구분하여 각 방법에 대한 일반적 전송 방식들을 두 가지 씩 4가지 타입을 제안하고, 각 타입의 일반적 처리 절차를 상태천이모델(Sate Transition Model)을 이용하여 설계한다. 도출된 각 상태천이도를 기반으로 각 타입들의 기능들을 실제로 구현하고 테스트함으로써 이들 방식들의 성능을 평가한다. 평가 결과로서, 다른 타입들과 비교해 볼 때, 여러 대기 상태들에서 데이터 혹은 응답 수신을 위해 저전력 Sleep모드-인터럽트와 폴링을 조합한 타입이 폴링만 사용하거나 Sleep모드-인터럽트만을 사용하는 방법보다 데이터 전송 실시간성, 데이터 유실도, 에너지 소모 등에서 가장 좋은 성능을 보인다.

주요어 : 사물인터넷, 센서네트워크, 게이트웨이

* This research was a part of the project titled "Development of the wide-band underwater mobile communication systems" funded by the Ministry of Oceans and Fisheries, Korea and this is also a little bit partly supported by "Full-time Graduate Student Support Work" from Graduate School of Gangneung-Wonju National University.

Received: 1 February 2016, **Revised**: 27 April 2016, **Accepted**: 15 July 2016

† **Corresponding Author**: Changhwa Kim
E-mail: kch@gwnu.ac.kr
Department of Computer Science&Engineering, Gangneung-wonju National University

1. 서론

사물인터넷(Internet of Things)을 구성하는 중요한 기반 구성요소 중의 하나는 유무선 통신/네트워크 인프라 구조로서 기존의 무선 모바일 망과 유무선 인터넷은 이 인프라 구조에서 가장 기반이 되는 중요한 필수요소임에는 더 말할 나위가 없다[1]. 이중에서 동일한 무선 통신/네트워크 프로토콜을 사용하는 동종의 무선 사물 서브 네트워크(이하 무선 사물 서브 망으로 칭함)를 무선 모바

일 망과 인터넷을 상호 연계하여 무선 데이터를 관제 사물로 전달하는 역할을 하는 무선 게이트웨이는 다수의 사물(노드)들로부터 많은 양의 센서 데이터를 모바일 망/인터넷을 통해 전달해야하기 때문에 그 기능과 성능이 무엇보다도 중요하게 되었다.

사물인터넷의 근간 기술인 USN (Ubiquitous Sensor Networks)이나 M2M(Machine-to-Machine) 기술의 일반적인 활용을 살펴보면 무선 사물 노드로부터 전송된 센서 데이터는 게이트웨이를 통해 모바일 망과 인터넷을 거쳐 관제 사물로 전달되는 과정에서 데이터의 유실 여부는 그다지 중요하게 취급되지 않았다. 그 이유는 활용 대부분이 모니터링을 중심으로 하는 분야들로서 데이터 유실이 목표 시스템이나 인간에게 영향을 전혀 주지 않거나 그 영향이 매우 미미해서 무시해도 되거나 혹은 모바일 망/인터넷을 100% 신뢰하는 전제로 개발되었기 때문이다. 또 한편으로는 송신 노드가 데이터 송신 후 목표 사이트로부터 데이터 수신 여부에 대한 응답 메시지 수신과 체크 그리고 데이터 재전송에서 수반되는 에너지 소모와 채널 성능 저하를 줄이고자 하는 이유이기도 하다.

이와는 반대로 원자력 발전소의 방사선 누출 감지, 가정이나 주변 환경의 화재 감지, 갑작스런 심장마비나 뇌졸증 등 생명에 대한 긴박성을 체크하는 U-헬스케어 등은 데이터의 유실이나 지연이 목표 시스템이나 인간에게 신체나 인명 혹은 재산상의 피해로 바로 이어지는 대표적인 분야들이다. 이 분야들은 사물이 송신한 데이터의 유실 여부에 대한 신속한 실시간 체크와 더불어 송신 데이터 유실로 판단되면 곧 바로 재전송하는 기능을 필요로 한다. 하지만 무선 사물 서브 망을 사용하는 각 사물 노드들이 이 기능을 직접 수행한다면 사물 노드가 송신한 데이터가 무선 사물 서브 망을 통해 게이트웨이에 안전하게 도착하였으나 모바일 망/인터넷의 문제로 인해 사물 노드가 송신한 데이터 유실로 인해 사물 노드가 같은 데이터를 재전송하면 무선 사물 서브 망과 게이트웨이를 통신경로를 다시 거쳐야한다. 이 경우 일반적으로 전송 속도가 느린 무선 사물 서브 망에서 재전송에 따른 긴박한 실시간 전송 지연 게이트웨이에 전달될 때까지 각 경로의 채널 점유로 인해 가용성 문제와 처리율에 대한 비효율성, 무선 사물 서브 넷 경로 상에서 사물 노드들은 라우팅으로 인한 추가적인 에너지 소모와 같은 문제들이 야기된다. 이 문제들은 송신 데이터가 게이트웨이에 안전하게 도착하면 그 후부터 타겟 노드의 데이터 수신 여부 체크와 재전송 기능을 사물 노드 대신 전송 메시지 응답 처리를 통하여 게이트웨이가 전담하면 해결된다.

본 논문에서 수행한 연구내용은 다음과 같다.

첫째로 무선 사물 서브 망을 모바일 망과 인터넷을 차례로 연계하는 무선 게이트웨이 처리 방식에 대해 앞에서 소개한 *전송 메시지 응답 무처리 기반의 게이트웨이*와 *전송 메시지 응답 처리 기반의 게이트웨이*로 구분하여 각 방법에 대한 일반적 전송 방식들을 2가지 씩 4가지 타입으로 제안한다.

둘째로 본 연구에서 제안한 각 타입의 일반적 처리 절차는 상태천이도(Sate Transition Diagram)를 이용하여 모델링한다. 상태천이도를 통해 모델링을 사용한 이유는 다음과 같은 장점을 가지기 때문이다.

- 모델링이 수월하다[2].
- 프로그램 설계가 수행되어 이벤트 중심의 프로그램 코드로 바로 변환이 가능하다.
- 각 타입들의 처리 방식을 비교하기 쉽다.
- 각 타입의 유지보수가 수월하다.

셋째로 본 논문은 도출된 각 상태천이도를 기반으로 각 타입들의 기능들을 실제로 구현하고 테스트함으로써 이들 방식들의 데이터 유실도, 실시간성, 그리고 에너지 소모 중심으로 성능을 비교한다. 각 타입들의 성능을 비교를 한 결과, 무선 사물 서브 망으로부터의 데이터 수신 대기 시 저 전력 Sleep 모드 상태와 Awake를 위한 인터럽트를 사용하고, 모바일 망 과 인터넷의 접속과 해지 시 각 접속 요청 혹은 해지 요청 후 응답 수신을 위한 폴링 대기 후 응답 내용에 따라 처리하는 방법이 데이터 전송 실시간성, 데이터 손실 방지, 에너지 효율성 측면에서 가장 높은 성능을 보임을 확인할 수 있었다.

본 논문의 내용 구성은 다음과 같다. 먼저 2절에서 관련연구를 소개하고, 3절에서는 무선 사물 서브 망을 모바일 망과 인터넷을 연결하는 게이트웨이의 일반적인 처리 방식을 4가지 타입으로 제안하고, 각 게이트웨이 처리 타입을 상태 천이 모델로 모델링하기 위한 구성요소들과 이벤트들을 소개한다. 4절에서는 현재 가장 일반적으로 사용되는 방법인 전송 메시지 응답 무처리 기반으로 하는 두 가지 타입의 게이트웨이 상태천이 모델을 소개하고, 5절에서 전송 메시지 응답 처리 기반으로 하는 두 가지 타입의 게이트웨이 상태천이 모델을 소개한다. 6절에서는 구현과 성능 테스트를 통하여 4가지 타입의 성능을 비교 분석을 제시한다. 마지막 7절에서는 본 논문의 의의와 향 후 연구방향을 제시하여 결론을 맺는다.

2. 관련연구

IoT을 비롯하여 USN, M2M 등에서 사용되는 게이트웨이는 MAC, Network, Application 등 여러 레벨에서 사용되고 있고, 그 중 일부는 모바일 망 또는 인터넷을 이용하여 인터넷에 연결을 통해 서비스를 제공하고 있다. 이 경우 IoT, USN, M2M 등에서 사용되는 게이트웨이는 통신매체로 Zigbee, Bluetooth, wifi 등을 이용한다[3]. 게이트웨이는 응용 계층에서의 게이트웨이 기법, DTN (Delay tolerant network)게이트웨이 기법 등이 이용되고 있다[4]. 응용 계층 게이트웨이는 자신이 필요한 정보만을 인터넷으로 전달해주는 전달 기능, 프런트 엔드 서버로 동작하여 네트워크의 다양한 정보를 미리 수집하여 데이터베이스에 저장을 한 뒤 인터넷 사용자가 필요에 의해 질의를 전송하면 해당하는 정보를 보내주는 등의 기능을 가지고 있다.

게이트웨이가 응용되어 사용된 예로 텔레매틱스가 있다[5, 6]. 텔레매틱스의 방법 중 하나로 AMI-C 표준 기반의 게이트웨이 구현에 대한 연구가 진행되었다. 이 방법은 차량 센서네트워크의 확장성에 초점을 두고 있으며, 자바기반으로 개발되어 이식성과 재사용성이 강한 특징을 지니고 있고, TCP/IP를 지원한다. 헬스케어 분야에서는 WBAN 게이트웨이 연구가 진행되었고 이 연구는 WBAN에서 측정한 혈압, 심전도, 산소포화도 등의 생체 데이터를 측정하여 인터넷을 통해 데이터베이스로 전송한다[7].

게이트웨이는 이종 네트워크의 통로 역할로 인해 통신이 자주 발생한다. 이러한 이유로 게이트웨이의 내부처리와 내부처리에 따른 에너지 소모가 중요하다. 그러나 앞서 소개한 연구에서는 게이트웨이 내부 처리방법 혹은 내부 처리방법에 따른 에너지 소모를 비교한 연구가 진행되지 않았다. 게이트웨이의 내부 처리방법에는 폴링방법과 인터럽트 방법이 있으며 인터럽트 방법에서는 명령에 대한 응답메시지를 무시하거나 폴링 혹은 인터럽트를 통해 수신한다.

그리고 통신 프로토콜이 복잡하거나 규모가 큰 프로그램을 코딩하는 경우 상태천이도를 이용하는 것이 한 방법이다. 예를 들어 프로토콜 구현을 위한 시간을 줄이기 위해 디자인패턴을 분석하고 상태천이도를 이용하여 설계하여 MiTS 통신 프로토콜을 구현한 연구가 진행되었으며 이 연구에서는 상태천이도를 이용하여 MiTS 통신 프로토콜을 구현하기위한 설계의 유연성, 모듈화, 재사용성, 이해도 등을 증가시켰고, 결과물의 신뢰성 또한 상향시켰다[2].

3. 게이트웨이 상태 천이 모델 구성요소

3.1 게이트웨이의 모바일 망/인터넷 연결 절차

무선 사물 서브 망과 모바일, 인터넷을 연결하는 게이트웨이를 포함하는 네트워크 시스템 아키텍쳐는 Fig. 1과 같이 사물인터넷을 구성하는 노드, 게이트웨이, 모바일 망, 인터넷, 서버로 구성된다. 게이트웨이의 망 연결 절차는 Fig.2와 같이 데이터 수신 단계, 모바일 망 접속 단계, 인터넷 접속단계, 데이터 전송단계, 인터넷 연결해지 단계, 모바일 망 연결해지 단계로 구성된다. 데이터 수신 단계는 게이트웨이의 무선 사물 서브 망의 노드(사물)로부터 데이터를 수신하는 과정이고, 모바일 망 접속 단계와 인터넷 접속단계는 게이트웨이의 모바일 통신 모듈을 통하여 모바일 망에 접속 후 인터넷에 접속하여 서버와 접속되는 과정이다. 이 두 접속이 완료되면 게이트웨이는 서버에 데이터를 올려줄 수 있고, 데이터가 모두 전송되면 게이트웨이는 차례로 인터넷과 접속을 끊고, 모바일 망과의 접속도 끊음으로서 모바일 망과 인터넷을 이용한 게이트웨이와 서버간의 데이터 전송이 종료된다.

Table 1. Gateway types by processing methods

Data Loss	Type	Data Reception Waiting Method	Waiting/Response/Response process in mobile network connection and disconnection	Waiting/Response/Response process in internet connection and disconnection
loss	타입-1	polling	delay/ignore/no	delay/ignore/no
	타입-2	interrupt	delay/ignore/no	delay/ignore/no
lossless	타입-3	interrupt	waiting/polling/yes	waiting/polling/yes
	타입-4	interrupt	Sleeping/interrupt/yes	Sleeping/interrupt/yes

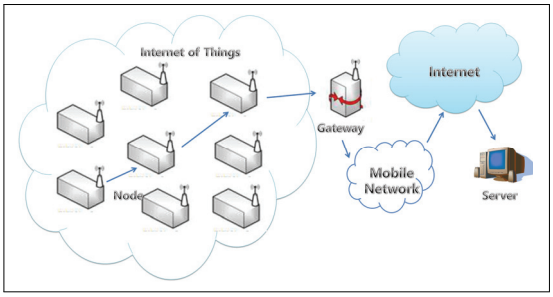

Fig. 1. Network system architecture using mobile network internet between a sub-network of things and a server

Fig. 2. Network connection process of the gateway

Table 2. States for state transition model

State	Meaning	State	Meaning
데이터_수신대기	게이트웨이가 어떤 노드로부터 데이터 수신을 기다리는 상태	데이터_전송대기	게이트웨이가 서버로 데이터 전송을 시도하는 상태
인터넷_접속해지대기 -Awake	게이트웨이와 서버와 연결 해지 시도에 대한 응답을 기다리는 상태(Awake 상태)	데이터_전송대기 -Awake	게이트웨이와 서버로 데이터 전송을 시도하고 응답을 기다리는 상태 (Awake 상태)
데이터_수신	게이트웨이가 어떤 노드로부터 데이터를 수신하는 상태	데이터_전송대기 -Sleep	게이트웨이와 서버로 데이터 전송을 시도하고 응답을 기다리는 상태(Sleep 상태)
데이터_수신_완료	게이트웨이가 어떤 노드로부터 마지막 데이터를 수신 완료한 상태	데이터_전송대기_완료	게이트웨이와 서버로 데이터 전송 대기완료 상태
모바일_망_접속대기	게이트웨이와 모바일 망과 접속을 시도하는 상태	인터넷_접속해지대기	게이트웨이가 인터넷과 연결해지를 시도하는 상태
모바일_망_접속대기 -Awake	게이트웨이와 모바일 망과 접속을 시도하고 응답을 기다리는 상태(Awake 상태)	모바일_망_접속대기 -Sleep	게이트웨이와 모바일 망과 접속을 시도하고 응답을 기다리는 상태 (Sleep 상태)
데이터_도착알림	게이트웨이가 어떤 노드로부터 데이터를 수신할 데이터가 있음을 알리는 상태	인터넷_접속해지대기 -Sleep	게이트웨이와 인터넷과 연결 해지 시도에 대한 응답을 기다리는 상태 (Sleep 상태)
모바일_망_접속대기_ 완료	게이트웨이와 모바일 망과 접속에 대한 대기완료 상태	인터넷_접속해지대기 완료	게이트웨이와 인터넷과 접속해지 대기완료 상태
인터넷_접속대기	게이트웨이와 인터넷과 접속을 시도하는 상태	모바일_접속해지대기_ 완료	게이트웨이와 모바일 망과 접속해지 대기완료 상태
인터넷_접속대기 -Sleep	게이트웨이와 인터넷과 접속을 시도하고 응답을 기다리는 상태(Sleep 상태)	모바일_망_접속완료_ 및_타임아웃시작	게이트웨이와 모바일 망과 접속을 완료하고, 타임인터럽트를 설정하는 상태
인터넷_접속대기 -Awake	게이트웨이와 인터넷과 접속을 시도하고 응답을 기다리는 상태(Awake 상태)	모바일_망_접속오류	게이트웨이와 모바일 망과 연결이 되어 있지 않은 상태에서 인터넷 과 연결을 시도하는 상태
인터넷_접속대기_완료	게이트웨이와 인터넷과 접속에 대한 대기완료 상태	모바일_망_접속해지 대기	게이트웨이가 모바일 망과 연결해지를 시도하는 상태
모바일_망_접속해지대 기 -Sleep	게이트웨이와 모바일 망과 연결 해지 시도에 대한 응답을 기다리는 상태(Sleep 상태)	모바일_망_접속해지 대기 -Awake	게이트웨이와 모바일 망과 연결 해지 시도에 대한 응답을 기다리는 상태(Awake 상태)
인터넷_접속완료_ 및_타임아웃시작	게이트웨이와 인터넷간의 접속이 완료되고, 타임인터럽트를 설정하는 상태	데이터_전송완료_ 및_타임아웃시작	데이터 전송을 완료하고 타임 인터럽트를 설정하는 상태
데이터_전송_실패	게이트웨이가 서버로 데이터 전송이 실패한 상태	인터넷_접속해지실패	게이트웨이와 인터넷과의 접속해지를 실패한 상태
인터넷_접속해지_완료	게이트웨이와 인터넷 간 접속해지를 완료한 상태	모든접속_해지	게이트웨이와 모바일 망, 인터넷 모두 접속 해지된 상태

3.2 단계별 처리방식에 따른 게이트웨이 타입 분류

게이트웨이가 3.1절에서 소개한 각 단계들을 어떻게 처리하느냐에 따라 일반적 처리 방법을 중심으로 Table 1.와 같이 4가지 타입으로 분류한다. Table 1.에서 타입-1과 타입-2는 각각 전송 메시지 응답 무처리 기반의 게이트웨이로 타입-1은 데이터 수신 시 데이터가 도착할 때까지 폴링하고 데이터 수신 후 모바일 망 접속과 인터넷 접속시 각각 MCU 타임 DELAY를 사용하여 대기하면서 응답수신을 무시함으로써 응답에 따라 전송, 재전송, 오류처리 등의 수행 없이 무조건 데이터를 전송하는 현재 자주 사용하는 일반적인 방법이다. 타입-2는 데이터 수신을 인터럽트 이벤트로 처리(즉, 외부로부터 데이터 혹은 메시지 도착 시 게이트웨이에 인터럽트 이벤트를 발생시켜 인터럽트 루틴을 실행하여 처리)하고 나머지는 타입-1과 동일하게 처리하는 방법이다. 반면, 타입-3과 타입-4는 전송 메시지 응답 처리 기반의 게이트웨이 데이터 수신은 타입-2와 같이 인터럽트 이벤트에 의해 처리한다. 이 두 방식의 차이점은 타입-3는 모바일 망과 인터넷 접속 시 폴링으로 대기하면서 수신한 응답메시지 내용에 따라 전송이나 재전송 혹은 오류를 처리하는 반면, 타입-4는 모바일 망과 인터넷 접속 시 Sleeping 모드에서 대기하면서 응답 수신 시 인터럽트 이벤트에 의해 Awake 하여 응답메시지 내용에 따라 데이터를 전송, 재전송, 혹은 오류를 처리하는 방식이다.

Table 3. Events for state transition model

Event	Menaing	Event	Meaning
TCP_WRITE(DATA)	게이트웨이가 서버로 데이터를 송신	TCP_OPEN(IP, PORT)	게이트웨이와 인터넷과 접속 요청
OUTER_ INTERRUPT	데이터 수신시 무선 사물 서브 망 통신모듈로 인해 게이트웨이 MCU가 인터럽트를 발생	DATA_REQUEST()	게이트웨이가 무선 사물 서브 망 통신모듈이 수신한 데이터 전송을 요청
REQUEST_SUCCESS	게이트웨이가 요청 된 내용의 성공을 알리는 메시지 수신	TCP_CLOSE()	게이트웨이가 인터넷과 접속해지 요청
RCV_DATA(D)	게이트웨이가 무선 사물 서브 망 통신모듈로부터 데이터를 수신하고 수신	LAST_DATA(D)=TRUE	게이트웨이가 무선 사물 서브 망 통신모듈로부터 마지막 데이터를 수신하고 수신을 종료
NET_CLOSE()	게이트웨이와 모바일 망과 접속해지 요청	WFI()	게이트웨이를 슬립모드로 전환
SOCKET_OPEN_ERROR	게이트웨이의 모바일 통신 모듈이 모바일 망과 인터넷을 거쳐 서버와 연결 시 소켓이 열리지 않는 이벤트 발생	CONNECTION_ TIMEOUT	일정시간동안 대기상태에서 머물다가 게이트웨이와 인터넷 혹은 게이트웨이와 모바일 망 의 모든 연결이 끊어진 경우
RECEIVE_RESPONSE_ TIMEOUT	게이트웨이가 요청에 대한 응답을 서버로부터 일정시간동안 받지 못하는 경우	DATA_SEND_OK	게이트웨이가 서버로 데이터 송신에 대한 응답메시지를 받은 경우
RESET()	게이트웨이 모바일 통신모듈을 재 부팅함	TIMEINTERRUPT_ DISABLE()	게이트웨이에서 타임인터럽트를 사용하지 않음으로 설정
TIMEINTERRUPT_ ABLE()	게이트웨이에서 타임인터럽트를 설정 혹은 재설정하여 카운트를 시작	DELAY(T)	게이트웨이가 시간 T동안 대기상태를 유지
NET_OPEN()	게이트웨이와 모바일 망과 접속 요청	DATA_RECEIVE_ CHECK()==FALSE	폴링 방법에서 센서네트워크 통신 모듈이 데이터를 미수신
DATA_RECEIVE _CHECK()==TRUE	폴링 방법에서 무선 사물 서브 망 통신모듈이 데이터를 수신		

3.3 게이트웨이 상태와 이벤트

본 논문에서 제안한 게이트웨이의 통신 방법을 상태천이 모델로 표현, 즉 설계하기 위해서는 이에 필요한 상태와 이벤트들을 식별하고 정의해야한다. 상태는 3.1절에서 소개한 게이트웨이의 처리과정은 통신 절차를 따라 데이터 수신, 모바일 망 접속, 인터넷 접속, 데이터 전송, 인터넷 접속해지, 모바일 망 접속해지와 관련하여 구성되며, 이벤트는 이들 상태상의 천이를 우발시키는 함수 호출과 이에 따른 에러 및 응답 수신 등으로 구성이 된다. 게이트웨이의 일반적 통신 방법을 상태천이모델로 표현하기 위한 상태들과 이벤트들을 각각 Table 2.와, 표 Table 3.에서 제안된다[8]

4. 전송 메시지 응답 무처리 기반의 게이트웨이

전송 메시지 응답 무처리 기반의 게이트웨이는 일반적으로 많이 쓰이는 방법으로, 3절에서 소개한 타입-1과 타

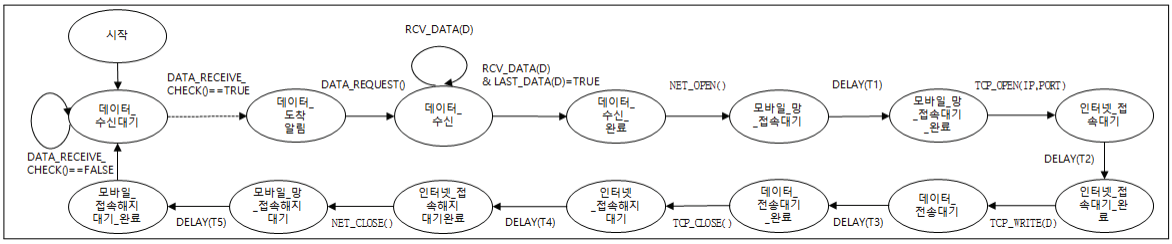

Fig. 3. Type-1 state transition model

```
Algorithm Type-1_Gateway(){
STEP 1. Send DATA_REQUEST();
STEP 2. state = 데이터_수신;
STEP 3. receivestate = 1;
        WHILE (1) {
          SWITCH (state) {
STEP 4.     CASE 데이터_수신대기 :
                IF DATA_RECEIVE_CHECK()==TRUE THEN
                  state = 데이터_도착_알림;
                ENDIF
                break;
STEP 5.     CASE 데이터_도착_알림 :
                Data_Request();
                state = 데이터_수신;
                break;
STEP 6.     CASE 데이터_수신 :
                D ← receive_data();
                IF LAST_DATA(D) == true THEN
                  state = 데이터_수신_완료;
                ENDIF
                break;
STEP 7.     CASE 데이터_수신_완료 :
                NET_OPEN();
                state = 모바일_망_접속대기;
                break;
STEP 8.     CASE 모바일_망_접속대기 :
                DELAY(T1);  // T1 = 시간(초)
                state = 모바일_망_접속대기_완료;
                break;
STEP 9.     CASE 모바일_망_접속대기_완료 :
                TCP_OPEN(IP, PORT); // IP = IP주소, PORT = PORT번호
                state = 인터넷_접속대기;
                break;
STEP 10.    CASE 인터넷_접속대기 :
                DELAY(T2);  // T2 = 시간(초)
                state = 인터넷_접속대기_완료;
                break;
STEP 11.    CASE 인터넷_접속대기_완료 :
                TCP_WRITE(D);
                state = 데이터_전송대기;
                break;
STEP 12.    CASE 데이터_전송_대기 :
                DELAY(T3) // T3 = 시간(초)
                state = 데이터_전송대기_완료 ;
                break;
STEP 13.    CASE 데이터_전송대기_완료 :
                TCP_CLOSE();
                state=인터넷_접속해지대기;
                break;
STEP 14.    CASE 인터넷_접속해지대기 :
                DELAY(T4) // T4 = 시간(초)
                state = 인터넷_접속해지대기_완료;
                break;
STEP 15.    CASE 인터넷_접속해지대기_완료 :
                NET_CLOSE();
                state = 모바일_망_접속해지대기;
                break;
STEP 16.    CASE 모바일_망_접속해지대기 :
                DELAY(T5) // T5 = 시간(초)
                state = 모바일_접속해지대기_완료;
                break;
STEP 17.    CASE 모바일_접속해지대기_완료 :
                state = 데이터수신대기상태;
                break;
          }
        }
```

Fig. 4. Algorithm transformed from type-1 sate transition model

입-2에 해당한다. 이 방법에 대한 상태천이도가 3.3절에서 식별정의한 상태와 이벤트를 이용하여 Fig. 3.과 Fig. 5와 같이 표현된다. 3.2절에서 소개한 타입 중 타입-1과 타입-2가 이 타입에 해당한다.·

4.1 타입–1 상태천이모델

타입-1을 나타내는 상태천이도는 Fig. 3.과 같다. 상태

천이도에서 게이트웨이의 내부 이벤트는 실선으로 외부 이벤트는 점선으로 외부 인터럽트는 일점쇄선으로 각각 표현한다. 타입-1은 앞서 소개한 상태천이도로 나타낸 Fig. 3.을 Fig. 4.와 같이 알고리즘으로 쉽게 변환할 수 있다. 상태천이도를 알고리즘으로의 변환에서 상태천이도의 각 상태는 CASE 문의 조건으로 표현되며, 현재 상태에서 이벤트에 의한 다른 상태로의 변이는 현재 상태의 CASE 문 조건 처리 부분에서 해당 이벤트 함수(혹은 명령)를 호출한 후 state 상태 변수를 변이하는 상태 값으로 변경함으로써 이루어진다. 상태 변이가 지속적으로 반복되므로 STEP 3에서 WHILE(1)로 처리되었음을 주시하자. 또한 이렇게 변경된 Fig. 4.와 같은 알고리즘의 CASE 문의 조건은 프로그램의 CASE로, 이벤트 함수 혹은 명령을 함수로 나타내어 쉽게 프로그램 구조로 자동생성이 가능하다. 또한 DELAY() 함수의 인자로 들어가는 T1, T2, T3, T4, T5 각 단계별로 게이트웨이가 서버로 명령을 보낼시 처리되는 최소 시간을 직접 측정하여, 사용하였고, 사용되는 게이트웨이의 모바일 망/인터넷 통신 명령을 수행할 수 있도록 수행 시간을 직접 측정하여 값을 구해야한다.

4.2 타입–2 상태천이모델

타입-2의 상태천이도는 Fig. 5.에 제시된다. 타입-2 게이트웨이는 앞서 소개한 타입-1과 거의 동일하나 데이터

수신대기 상태에서 평상시에는 Sleep 상태에 있다가 무선 사물 서브 망 통신모듈이 데이터 수신을 알려주는 인터럽트 이벤트(OUTER_INTERRUPT)를 발생시키면 깨어나서(Awake) 데이터도착알림 상태로 변이하는 차이점을 갖는다. 이 때문에 데이터 수신 대기 시간이 길어지면 길어질수록 타입-2는 타입-1에 비해 에너지 소모를 더욱 줄일 수 있다.

4.3 타입–1와 타입–2의 문제점

타입-1과 타입-2 상태천이모델을 보면 모바일 망 및 인터넷 접속과 해지를 위한 이벤트들에서 수반되는 다양한 에러 처리 이벤트와 이에 따른 변이가 없기 매문에 처리 방법이 매우 단순한 장점을 갖는다. 그러나 타입-1과 타입-2는 다음에 설명할 몇 가지 문제점들을 갖는다.

첫째로는 게이트웨이가 데이터수신대기 상태에서 하나의 데이터를 수신한 후 모바일 망 접속, 인터넷 접속, 데이터 송신, 망 해제를 거쳐 다음 번 데이터 수신을 위해

다시 데이터수신대기 상태로 진입할 때까지의 송신 사이클 중에 게이트웨이 MCU는 busy 상태가 되는데, 이 때 게이트웨이 무선 사물 서브 망 통신모듈에 수신 큐가 존재하지 않으면 busy 상태에서 다른 데이터 수신은 불가능하므로 데이터 유실이 발생한다. 반면에 수신 큐가 존재할 경우 무선 사물 서브 망 통신모듈은 수신되는 데이터들을 차례로 큐에 저장하면서 게이트웨이 MCU가 진행 중인 송신 사이클 완료 후 다시 데이터수신대기 상태로 돌아 올 때까지 데이터 수신 알림을 지연시킬 수 있어 수신 데이터의 유실을 일부 막을 수 있다. 하지만 송신 사이클 처리 속도보다 데이터 수신 시간이 더 빠르면 궁극적으로 수신 큐가 오버플로우 상태가 되어 수신 데이터 유실이 발생한다.

둘째로, 타입-1과 타입-2에서 사용하는 DELAY(T)(여기에서, T는 대기시간) 이벤트는 모바일 망이나 인터넷 접속 혹은 해지 요청 후에 수신되는 응답을 무시하기 위해 T 동안 MCU busy 상태의 대기를 발생시키는데 이를 위해서는 응답이 대기시간 T 이내에 이루어져야하기 때문에 대기시간 T는 응답수신 시간보다 더 긴 시간을 요구한다. 이로 인해 데이터 송신 사이클이 더 길어져 수신

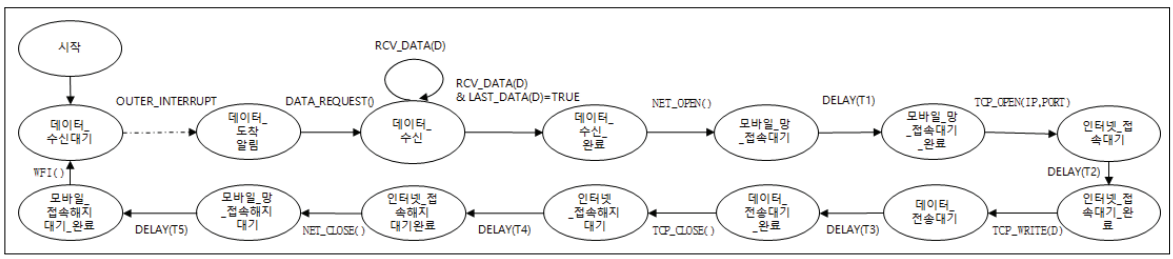

Fig. 5. Type-2 state transition model

데이터의 유실 확률이 더 커지게 된다.

셋째로, 타입-1과 타입-2는 모바일_망_접속대기 상태와 인터넷_접속대기 상태에서 응답 대기 시 DELAY(T)를 사용하여 응답 수신을 무시하기 때문에 예외가 발생하더라도 예외처리 없이 다음 상태로 진입하게 되는데 사실 상 이들 중 한 상태라도 예외가 발생하면 망 접속에 실패한 것이므로 데이터 전송이 불가능하여 데이터 유실이 발생하게 된다.

마지막으로 타입-1은 송신 사이클 내내 MCU-bound 상태이기 때문에 전력 소모가 커지며 타입-2도 데이터 수신 시간 간격이 짧으면 나머지 사이클 부분이 MCU-bound이므로 이 역시 전력 소모가 커지게 된다.

5. 전송 메시지 응답 처리 기반의 게이트웨이

4.3절에서 이미 소개했듯이 타입-1과 타입-2는 송신 사이클 전체 혹은 대부분이 MCU-bound이기 때문에 수신 데이터 유실 가능성과 전력 소모가 커지는 문제점이 존재한다. 또한, 망 접속 대기 상태에서 망 불안정성에 따른 에러 등의 응답에 대한 예외 처리가 없으므로 예외 발생 시 송신 데이터 유실이 발생한다. 그리고 각 망 접속 대기 상태에서 부적합한 명령을 주는 경우 상태에러가 발생 할 수 있으며, 상태 에러 발생 시 후속적으로 시도되는 데이터 송신은 불가능하다. 게다가, 모바일_망_접속완료 상태 혹은 인터넷_접속완료 상태 이후 데이터 전송이 이루어지지 않으면 통신사(예 : SKTelecom)의 망 정책에 의해 일정 시간 이후 모바일 망이나 인터넷 접속을 일방적으로 해지하기도 하며 이 때 데이터 전송은 물론 상태 에러가 발생하기도 한다. 이러한 문제점을 해결하기 위해 이 절에서는 외부 인터럽트와 예외 처리를 수행하는 타입-3와 타입-4를 소개한다.

5.1 타입–3 상태천이모델

타입-3의 상태천이도가 Fig. 6.에서 제시된다. 타입-3는 각 상태 혹은 이벤트 처리 중에도 데이터 수신 외부 인터럽트 이벤트 발생 시 이 인터럽트요청이 실시간으로 인터럽트 큐에 저장되도록 하고, 진행 중인 데이터 송신 사이클이 끝나서 데이터_수신대기 상태가 되면 큐에 대기하고 있는 다음 번 데이터 수신 인터럽트 이벤트 요청 처리와 함께 다음 사이클을 시작하도록 모델링 되었다. 이로 인해 MCU가 busy 상태에서도 수신 데이터가 유실되지 않는 효과를 얻을 수 있다. 이 모델은 모든 대기 상태(즉, 모바일_망_접속대기상태와 인터넷_접속대기 상태, 데이터_전송_대기상태, 인터넷_접속해지 대기상태, 모바일망_접속해지_대기 상태)에서 응답 수신을 MCU-bound로 체킹하고 처리하는 방법을 사용한다. 따라서 외부로부터 응답 수신이 발생하자마자 바로 수신 및 처리할 수 있게 되어 데이터 송신 사이클 시간이 최소화되며 이로 인해 에너지 절약 효과를 얻을 수 있다.

또한, 타입-3는 게이트웨이가 모바일_망_접속대기 상태, 인터넷_접속대기 상태, 데이터_전송대기 상태에서 응답 내용에 따른 상태 변이가 이루어지도록 했으며, 응답 메시지를 받지 못할 경우 응답 시간제한(RECEIVE_RESPONSE_TIMEOUT)을 두어 이 시간이 만료되면 각 대기 상태에서 빠져나와 연결을 재시도 하도록 모바일_망_접속실패 상태, 인터넷_접속실패 상태, 데이터_전송

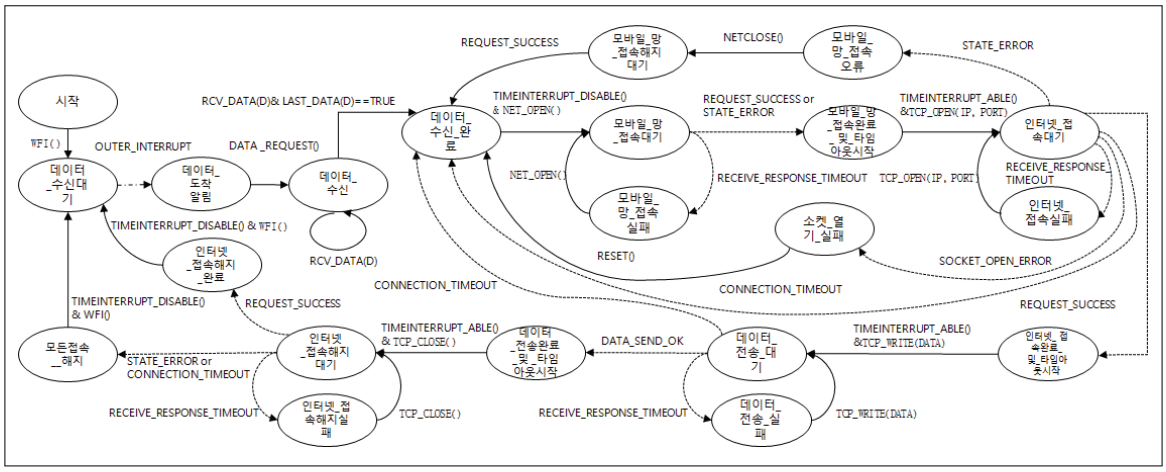

Fig. 6. Type-3 state transition model

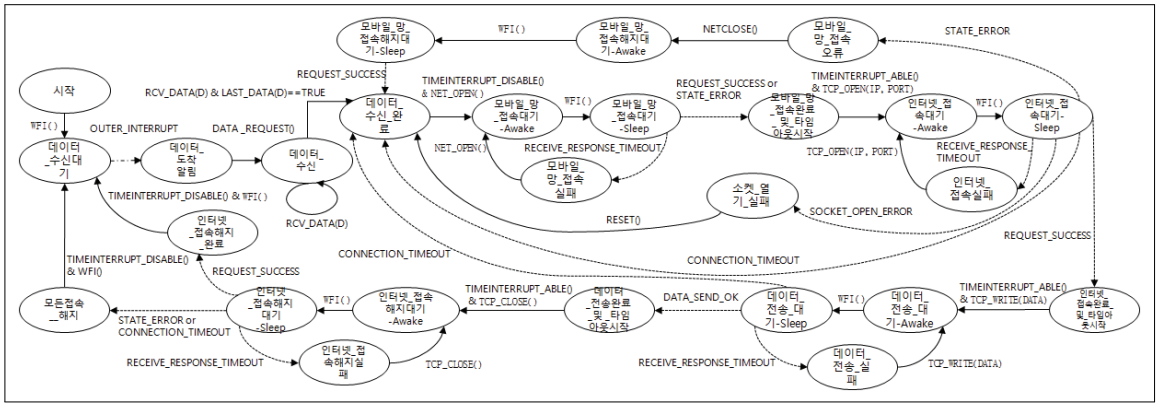

Fig. 7. Type-4 State Transition Model

실패 상태를 새롭게 소개함으로써 망 접속 실패로 인한 데이터 유실 문제를 해결하였다. 모바일_망_접속완료_및 _타임아웃시작 상태와 인터넷_접속완료_ 및_타임아웃시작 상태, 데이터_전송완료_및_타임아웃시작 상태에서 각각 응답 제한시간을 초기화하여 시작하도록 모델링 되었다. 또한, 모든 접속_해지 상태와 인터넷_접속_해지_완료 상태, 데이터_수신_완료 상태에서는 TIMEINTERRUPT_DISABLE() 이벤트를 발생시켜 이전 상태에서 시작한 응답 시간제한을 중단하도록 모델링 했다. 게이트웨이가 모바일 망 혹은 인터넷 접속 후 데이터 전송이 없거나 혹은 인터넷 재연결을 계속 시도하거나 송신하지 못한 데이터 전송을 계속 시도하는 경우, 또는 게이트웨이가 인터넷과 접속해지를 계속 시도하는 경우 일정 시간(CONNECTION_TIMEOUT)이 지나면 앞에서 언급했듯이 통신사 망 정책에 의해 일방적으로 외부에 의해 모든 연결이 끊어지기 전에 모바일 망과 다시 연결하는 과정이 필요하다. 이 경우는 인터넷_접속대기 상태, 데이터_전송대기 상태에서 발생할 수 있으며, 이 문제를 해결을 위해 이들 각 대기상태에서 일정 주기마다 타임 인터럽트 이벤트(CONNECTION_TIMEOUT 이벤트)를 발생시켜서 모바일 망 재접속을 위해 데이터_수신_완료 상태로 전이하도록 함으로써 외부에 의한 모바일 망해지 오류로 인한 데이터 유실을 방지했다.

5.2 타입-4 상태천이모델

타입-4에 대한 상태천이모델은 Fig. 7.에서 제시된다. 타입-4와 타입-3의 근본적 차이점은 모든 대기 상태에서 타입-3는 응답 수신을 MCU-bound로 체킹하는 반면, 타입-4는 Sleep 모드로 전환한 후 데이터 혹은 응답 수신 인터럽트 이벤트에 의해 Awake하여 수신 메시지 내용에 따라 다음 상태로 변이하는데 존재한다. 그 외의 나머지 상태와 이벤트 그리고 천이 관계는 타입-3과 같다.

Table. 4. Gateway specifications

H/W Parts	Specifications
MCU	Cortex-M4
Sub-Network Module	CC2530
Mobile Internet Module	RCU-890

Table. 5. Thing node specifications

H/W Parts	**Specifications**
MCU	Cortex-M3
Sub-Network Module	CC2530

6. 성능실험

4절과 5절에서 제시한 4가지 게이트웨이 타입들은 상태천이모델 기반으로 변환된 알고리즘을 쉽게 코딩한 후 데이터 유실과 에너지 소모 비교를 위해 성능 테스트를 수행했다. 이를 위해 사용한 게이트웨이 H/W 사양은 Table. 4.와 같고, 사물 노드의 사양은 Table. 5.와 같다.

서버는 TCP/IP서버를 이용하였다. Fig. 8.과 같이 게이트웨이가 모바일 망/인터넷 접속을 위해 CDMA 통신모듈 RCU-890을 이용하였고, 게이트웨이와 무선 사물 서브 망의 노드간의 통신은 무선 사물 서브 망 통신모듈 CC25 30을 이용하였다. 게이트웨이와 무선 사물 서브 망 통신모듈은 동일한 무선 전원으로 공급하고 CDMA 통신 모듈은 별도의 전원을 공급하였다. 무선 사물 서브 망 통신모듈 CC2530의 경우 수신 큐와 송신 큐를 가지

Table 6. Data loss according to error types

Error Types	Type-1	Type-2	Type-3	Type-4
SOCKET_OPEN_ERROR	loss	loss	lossless	lossless
CONNECTION _TIMEOUT	loss	loss	lossless	lossless
RECEIVE_RESPONSE _TIMEOUT	loss	loss	lossless	lossless

고 있어 무선 사물 서브 망 통신모듈이 다른 무선 사물 서브 망 통신모듈 CC2530로 보내야 하는 데이터가 한 개 이상 쌓이면 송신 큐에 저장하고, 다른 무선 사물 서브 망 통신모듈 CC2530로부터 수신하여 게이트웨이 MCU가 수신해야할 데이터가 한 개 이상 쌓이면 수신 큐에 저장한다.

6.1 예외 처리 테스트와 데이터 유실

테스트를 통해 각 게이트웨이 타입에 따른 데이터 유실 여부 결과가 Table 6.에 제시되었다. 게이트웨이와 서

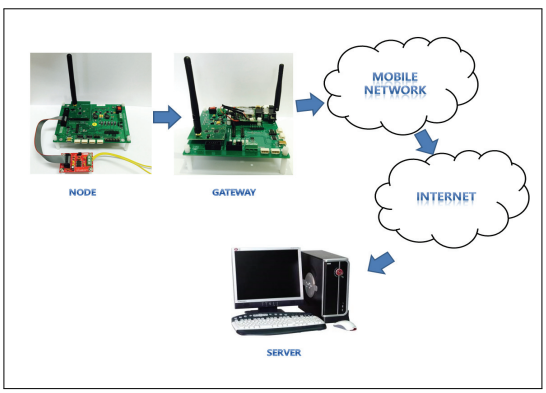

Fig. 8. Network environment for experiments

버간의 접속을 시도 중 통신환경이 좋지 않아, 소켓을 열지못하는 경우에 SOCKET_OPEN_ERROR가 발생한다. SOCKET_OPEN_ERROR가 발생하는 환경은 서버를 종료시킨 후, 게이트웨이와 서버간의 접속을 계속 시도하여 조성했다. SOCKET_OPEN_ERROR가 발생한 경우 타입-1 게이트웨이와 타입-2 게이트웨이는 서버로의 데이터 유실이 발생하는 반면, 타입-3와 타입-4인 경우 게이트웨이의 CDMA는 모바일 망/인터넷 재접속 과정을 거쳐 데이터를 송신한다. CONNECTION_TIMEOUT의 경우 게이트웨이가 인터넷과 연결을 계속 시도하거나 데이터 전송 시도 혹은 인터넷과 연결 해지를 계속 시도하거나 데이터 전송 시도 혹은 인터넷 망과 연결 해지를 계속 시도하는 경우 일정 시간이 지나면 통신사에서 모든 연결을 끊어버리는 절차 때문에 생기는 에러로 이 경우 처음 과정인 모바일 망과 접속부터 인터넷 접속 및 서버로 데이터 송신까지의 절차를 행하게 된다. 응답 제한시간(RECEIVE_RESPONSE_TIMEOUT) 경과 예외처리 테스트의 경우 CDMA의 안테나를 뽑거나 서버를 종료함으로써 이 오류 상태 환경을 조성할 수 있다. 이 경우 타입-3와 타입-4는 모바일 망 혹은 인터넷에 계속 연결을 시도하도록 되어있고, 다시 통신환경이 좋아지면 게이트웨이가 모바일 망 혹은 인터넷과 연결이 된다.

6.2 데이터 수신 주기에 따른 실시간성과 데이터 유실 정도

데이터 수신 시간 간격에 따른 데이터 유실 정도를 실험하기 위해 본 연구에서는 무선 사물 서브 넷의 사물 노드가 각 타입의 게이트웨이로 30분 동안 1초, 3초, 6초,9초 주기로 데이터를 송신하도록 하고 게이트웨이는 이 데이터를 모바일 망과 인터넷을 통해 서버로 데이터를 송신하도록 함으로써 서버에서 수신된 데이터 개수를 카운트함으로써 측정하였다. 시간 간격을 1초, 3초, 6초, 9초 주기로 잡은 이유는 다수의 사물들이 동시에 경쟁적으로 게이트웨이로 데이터를 송신하는 극한 상황에서 각 타입의 게이트웨이 타입어느 정도의 데이터 유실을 보이는지를 보이기 위해서이다,

이 성능 실험 결과, Fig. 9.에서 무손실 데이터를 백색으로 손실 데이터를 회색으로 표기하였고, 타입-3와 타입-4는 사물 노드가 전송한 데이터를 모두 수신하여 서버로 데이터를 실시간으로 전송시 0% 데이터손실이 있는 반면, 타입-1은 시간 지연과 함께 1초 주기에서 96.7%(Fig. 9. (a)), 3초 주기에서 83.7%(Fig. 9. (b)), 6초 주기에서 66%(Fig. 9. (c)), 9초 주기에서 51%(Fig. 9. (d))의 데이터 손실을 보였고, 타입-2는 역시 시간 지연과 함께 1초 주기에서 96.8%(Fig. 9. (a)), 3초 주기에서 83.5%(Fig. 9. (b)), 6초 주기에서 66.4%(Fig. 9. (c)), 9초 주기에서 51%(Fig. 9. (d))의 데이터 손실을 보였다. 타입-1과 타입

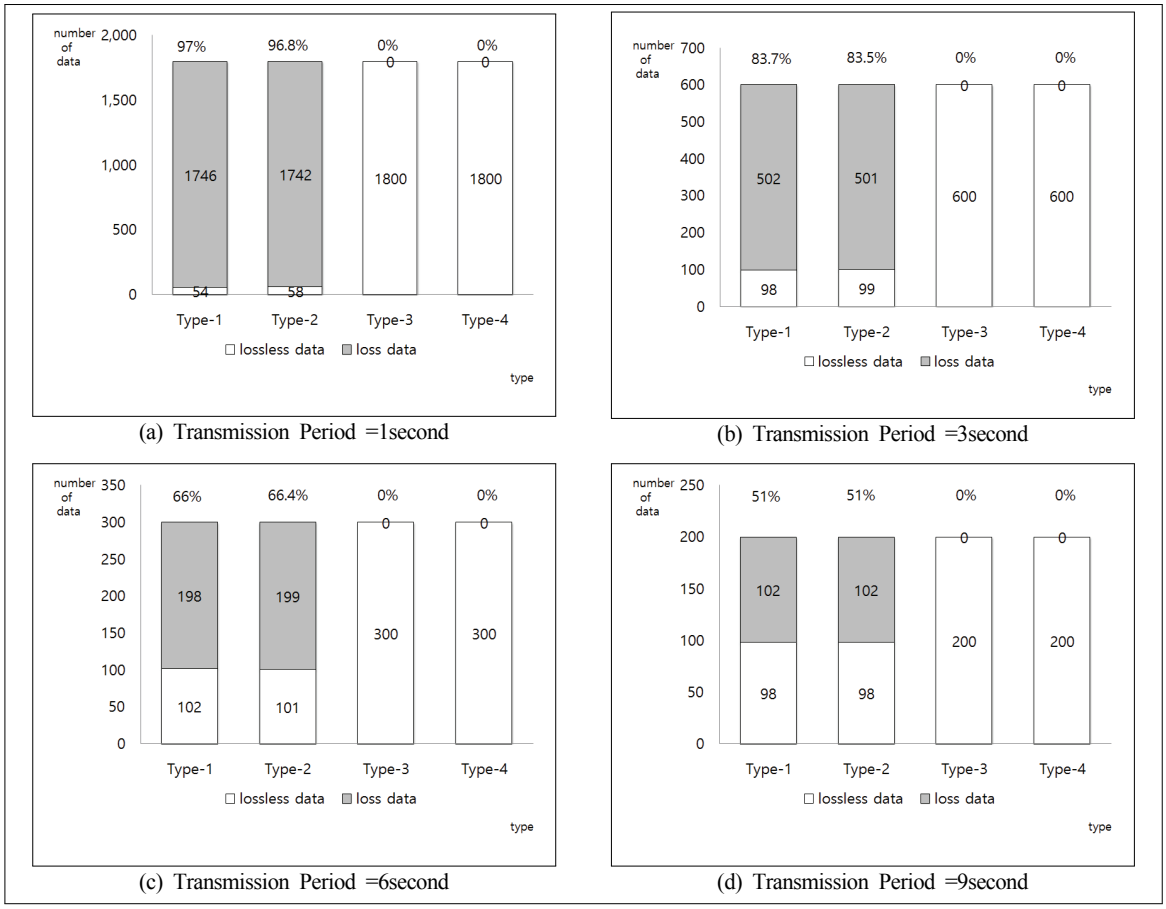

Fig. 9. Data loss rates according to data transmission periods

-2가 1초 주기에서 시간 지연이 발생하고 데이터 수신율이 매우 낮은 이유는 모바일 망과 인터넷의 접속과 해지 시 응답 수신을 무시하기 위해 고정된 DELAY 처리 때문인데, 이로 인해 게이트웨이의 데이터 송신 주기가 길어지고 이 주기 동안에는 데이터 수신 처리를 못하기 때문이다. 사물 노드와 게이트웨이는 각각의 무선 사물 서브 망 통신모듈(CC2530)에 데이터 송신 큐와 수신 큐를 가지고 있다. 하지만 1초 주기 송신의 경우 게이트웨이의 데이터 수신 주기가 데이터 송신 주기보다 매우 짧아서 먼저 게이트웨이 통신모듈 수신 큐가 다 차게 되는데, 그 이후 사물 노드는 게이트웨이로의 송신 지연이 발생해서 송신할 데이터를 사물 노드의 통신모듈 송신 큐 저장하다가 오버플로우가 발생하여 예외 처리를 못하게 됨으로써 그 이후에 송신하려는 데이터가 모두 유실되는 현상이 발생했기 때문이다. 타입-1과 타입-2의 3초, 6초, 9초, 12초 주기에서도 역시 시간 지연이 발생함과 동시에 1초 주기처럼 짧은 시간에 오버플로우가 일어나지 않지만 어느 정도 시간이 지남에 따라 같은 이유로 송신 데이터의 유실이 발생한다. 결론적으로 타입-3와 타입-4은 데이터 무유실율과 실시간성에 대한 성능이 타입-1과 타입-2에 비해 높은 것을 알 수 있다.

6.3 데이터 수신 주기에 따른 에너지 소비 성능 비교

무선 사물 서브 망으로부터의 데이터 수신과 각 게이트웨이 알고리즘 처리에 따른 에너지 소비를 서로 비교하기 위해 12V 건전지로 무선 사물 서브 망 통신모듈과 게이트웨이 보드에 전원을 공급하였고, 모바일 인터넷 통신모듈(CDMA, RCU-890)은 별도의 외부 전원을 사용했다. 실험에서 3분 주기, 6분 주기, 9분 주기로 사물 노드가 게이트웨이로 데이터를 송신하도록 하였고 전원 공급 건전지가 소모되어 게이트웨이가 모바일 망과 인터넷으로 데이터를 더 이상 전송하지 못해 결국 서버가 데이터

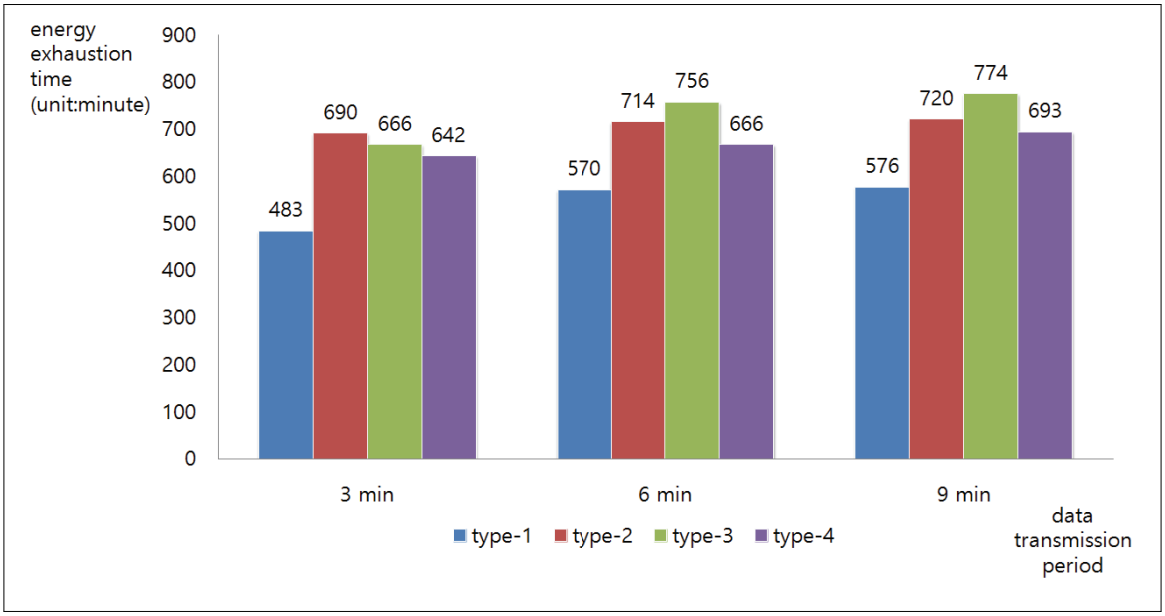

Fig. 10. Energy exhaustion time durations in data transmission periods

를 받지 못하는 시간까지 실험한 후, 서버가 수신한 데이터 개수와 지속시간을 측정하여 에너지 소비 성능을 비교했으며, 그 결과가 Fig. 10.에 제시되었다.

3분 주기인 경우 데이터를 타입-2, 타입-3, 타입-4, 타입-1의 순서로 에너지 소비량이 적었고, 6분 주기 그리고 9분 주기에서는 타입-3, 타입-2, 타입-4, 타입-1의 순서를 보였다. 3분 주기에서 타입-1이 다른 타입들에 비해 에너지 소모가 많은데 그 이유는 달리 타입-1을 제외한 나머지 타입들이 데이터 수신 대기 상태에서 Sleep 절전 모드에 있기 때문에 타입-1보다 전력 소모가 적게 되는 것으로 분석된다. 6분 주기와 9분 주기에서 타입-3가 다른 타입들에 비해 에너지 소모가 적은 이유는 타입-1의 경우 3분 주기에서 타입-2가 타입-1보다 에너지 소모가 적은 이유와 동일하다. 한편, 3분 주기의 결과와는 다르게 타입-3가 타입-2보다 에너지 소모가 적은 이유는 타입-2의 경우 DELAY 시간이 최대시간으로 잡혀있기 때문에 타입-3에 비해 타입-2가 CPU 가동시간이 길고, 게이트웨이가 데이터 보내는 주기가 길어져 응답 처리의 발생빈도가 줄어서 이에 상응하는 에너지 소모가 줄기 때문이며, 타입-4보다 타입-3가 에너지 소모가 적은 이유는 타입-4 경우 각 대기 상태의 Sleep 모드에서 응답 수신에 따른 인터럽트 루틴 처리 오버헤드가 더 높은 에너지 소모를 발생시킨 것으로 분석된다.

6.1절부터 6.3절까지의 성능 분석을 종합하면 타입-3가 데이터 유실성, 실시간성, 에너지 효율성에서 가장 좋은 것으로 평가되며, 모든 대기 상태를 MCU-bound로 처리하는 타입-1의 경우 세 가지 성능 모두 가장 나쁜 것으로 평가된다. 한편, 타입-3와 타입-4를 비교할 때 대기 상태에서 Sleep 절전 모드가 항상 에너지 효율성을 보장하는 것은 아니며, 대기상태 시간이 짧아짐에 따라 응답 처리를 위한 인터럽트 이벤트 처리 오버헤드가 에너지 소모를 발생시켜 에너지 효율성을 오히려 더 낮추는 결과를 초래함을 알 수 있다.

7. 결론 및 향후연구과제

본 논문에서는 게이트웨이를 폴링방법을 이용하여 데이터 전송 혹은 서버와 연결에 대한 응답을 DELAY로 무시하는 타입-1과 타입-1에서 폴링방법에서 인터럽트 방법으로 개선한 타입-2, 타입-2처럼 인터럽트 방법이지만 데이터 전송 혹은 서버와 연결에 대한 응답을 인터럽트 내에서 폴링 방법을 수신하여 처리하는 타입-3, 데이터 전송 혹은 서버와 연결에 대한 응답을 인터럽트로 수신하는 총 4개의 타입으로 나누어서 상태천이모델을 통해 프로그램을 설계하고 구현하여 각각의 타입을 데이터 유실, 실시간성, 에너지 소비에 대해 실험을 하였다.

본 논문에서 4가지 타입으로 나누어서 실험 결과 타입-3가 데이터 유실, 실시간성, 에너지 소비 항목 모두에서

가장 좋았고, 그에 반해 타입-1은 세 항목에서 모두 좋지 못다는 사실을 알 수 있었고, U-헬스나 원자력발전소와 같이 생명이나 생태계와 관련되어 하나하나의 데이터가 중요한 경우 타입-3 방법의 사용을 제안하고, 일반적인 모니터링에서 하나하나의 데이터가 중요하지 않은 경우에는 다른 타입의 방법을 사용해도 된다는 방향성을 제시해 줄 수 있는 것에 의의가 있다.

향후 연구로는 게이트웨이를 단방향 통신 방법을 양방향 통신으로 개선하여 데이터 유실, 실시간성, 에너지소비에 좋은 방법을 연구하는 것이다.

References

1. Min, K.S., Internet of Things, NET Term, 2014.
2. Park, H.C., J.S. Lee, K.W. Jang, “An Objected-Oriented Design and C++ Implementation of MiTs Communication Protocol Specified in State Transitions”, *Journal of the Korean Society of Marine Engineering*, 2010, Vol. 34, No. 1, pp. 147-153.
3. Davidson, I., Machine-to-Machine(M2M) Gateway: Tru sted and Connected Intelligence, freescale, 2013
4. Pack, S.H., M. Zhang, D.H. Chang, K.D. Cho, Y.H. Choi, T.Y. Kwon, “A Survey on Integration Schemes of Wireless Sensor Networks and Internet for Ubiquious Sensor Networks”, Telecommuni cations Review, 2005, Vol. 15, No. 2, pp. 337-350.
5. Park, P.S., H.B. Yim, Y.S. Won, J.E. Jung, “Implementation of the In-Vehicle Smart Sensor Network Gateway”, The *Korean Society Of Automotive Engineers*, 2007, pp. 52-57.
6. Won, Y.S., H.B. Yim, P.S. Park, J.E. Jung, “Implementation of AMI-C based Sensor Network Gateway Platform”, The *Korean Society Of Automotive Engineers,* 2007, pp. 35-40.
7. Park, Y.M., “A Study on the Implementation of WBAN-Based Medical Gateway”, *2015 Summer KICS,* 2015, pp. 1386-1387.
8. Seong, C.J., C.H. Kim, "Identification of Components for State Transition Model-based Design of Wireless Gateways to Connect between a Sub-network and Mobile Internet", The 2016 Spring Conference *of the Korea Multimedia Society,* 2016, pp. 319-322.

성 철 제 (Cheol-Je SEONG)

2013년 2월 : 강릉원주대학교
컴퓨터공학과(공학사)
2013년 9월~현재 : 강릉원주대학교
컴퓨터공학과 공학석사

관심분야 : Internet of Things, Sensor Network, Security

김 창 화 (Changhwa KIM)

1985년 2월 : 고려대학교
수학교육과(이학사)
1987년 2월 : 고려대학교
전산학전공(이학석사)
1990년 2월 : 고려대학교
전산학전공(이학박사)

1994년~1995년 : University of Toronto, Enterprise Integration Lab. Post-Doc. & Visiting Professor
2002년~2004년 : 미국 Texas A&M대학 VIsiting Scholar
1989년~현재 : 강릉원주대학교 컴퓨터공학과 교수
2005년~현재 : 강릉원주대학교ITRC(해양센서네트워크시스템기술연구센터) 센터장

관심분야 : Internet of Things, Distributed System, Intelligent System, Underwater Communication and Sensor Network