

지오센서 네트워크에서 에너지 효율성을 지원하는 바이트코드 기반 데이터 서비스 미들웨어 구현

Implementation of Bytecode based Data Service Middleware Supporting Energy Efficiency in Geosensor Networks

홍 승 태* 윤 민** 장 재 우***
Seung Tae Hong Min Yoon Jae Woo Chang

요약 최근 모바일 기기 및 무선 통신의 발달로 인하여 지오센서 네트워크가 실생활의 다양한 분야에서 활용되고 있다. 이에 따라 센서 노드의 제한된 에너지를 효율적으로 사용할 수 있는 미들웨어에 대한 연구가 활발히 진행되고 있다. 그러나 기존 미들웨어들은 센서 노드의 고유한 처리 능력 및 기능을 고려하여 개발하지 않았기 때문에, 제한된 시스템 자원만을 사용하는 센서 노드에서는 운영이 불가능하다. 따라서 본 논문에서는 지오센서 네트워크에서 에너지 효율성을 지원하는 바이트코드 기반 데이터 서비스 미들웨어를 설계 및 구현한다. 제안하는 미들웨어는 첫째, 하드웨어 추상화를 지원하는 데이터 관리자와 최소한의 바이트코드 명령어를 통하여, 센서 노드에 최적화된 기능만을 제공한다. 둘째, 불필요한 데이터를 제거함으로써 데이터 전송량을 최소화하는 데이터 집계 질의처리 및 데이터 필터링을 통하여, 센서 노드의 에너지 효율성을 향상시킨다. 마지막으로, 기존 연구인 SwissQM과의 성능 평가를 통해 제안하는 미들웨어가 데이터 전송량과 평균 에너지 소모량 측면에서 에너지 효율적임을 보인다.

키워드 : 지오 센서 네트워크, 데이터 서비스 미들웨어, 에너지 효율성

Abstract Recent development in wireless communication and mobile positioning technologies make geosensor networks widely used in the various fields of real world. As a result, much research has been done on the middleware that uses limited energy resources efficiently. However, because traditional middleware does not consider the characteristics of sensor node, such as computing power and specification, the existing middleware can not support the sensor nodes with only the restricted system resource. Therefore, in this paper, we design and implement a new Bytecode based Data Service Middleware supporting energy efficiency in geosensor networks. At first, the proposed middleware provides the optimized functions for sensor nodes by using minimum bytecode instruction set and data manager supporting hardware abstraction. Secondly, the proposed middleware increases the energy efficiency of sensor node through both data aggregation query processing and data filtering that minimize data transmission by eliminating unnecessary data. Finally, we show from our performance analysis that the proposed middleware is more energy efficient than the existing SwissQM.

Keywords : Geosensor Networks, Data Service Middleware, Energy Efficiency

1. 서론

최근 유무선 통신 기술의 발전 및 모바일 정보기기의 보편화에 힘입어, 시간과 장소에 제약 없이 지리공간상에서 발생하는 다양한 현상들을 제공할 수

있는 지오센서(geosensor) 네트워크 기술이 각광받고 있다.[1] 지오센서 네트워크란 지리공간상에서 발생하는 다양한 현상들을 모니터링 하는 특정형태의 센서 네트워크 인프라 및 관련 소프트웨어를 의미한다.[2] 지오센서 네트워크는 환경/농업, 시설물

† 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.

* 전북대학교 컴퓨터공학과 박사과정 sthong@dblab.chonbuk.ac.kr

** 전북대학교 컴퓨터공학과 석사 myoon@dblab.chonbuk.ac.kr

*** 전북대학교 컴퓨터공학과 교수 jwchang@chonbuk.ac.kr(교신저자)

관리, 기상관측, 헬스 케어 등 다양한 분야에서 활용되고 있으며, 최근 이를 이용한 대규모의 프로젝트가 진행되고 있다.

한편, 지오센서 네트워크의 하드웨어 및 미들웨어 기술 중에서 하드웨어에 대한 연구는 활발히 진행되어 왔으나, 미들웨어에 대한 연구는 아직 초기단계에 머물러 있다. 이는 기존 데이터 서비스 시스템과 같은 데이터 서비스 미들웨어는 센서 노드의 고유한 처리 능력 및 기능을 고려하여 개발하지 않았기 때문에, 제한된 시스템 자원만을 사용하는 센서 노드에서는 운영이 불가능하기 때문이다. 따라서 센서 노드의 제한된 에너지를 효율적으로 사용하기 위해, 지오센서 네트워크에서 에너지 효율성을 지원하는 데이터 서비스 미들웨어에 대한 연구가 필수적이다.

데이터 서비스 미들웨어는 지오센서 네트워크에서 발생한 이벤트들의 효율적인 처리 및 관리를 위한 기술로써, 네트워크 응용 서비스 계층과 센서 노드 하드웨어 계층의 중간에 위치하여 이들 간의 유기적인 연동을 지원한다. 대표적인 미들웨어인 TinyDB[3]는 SQL과 유사한 고수준의 질의어를 지원함으로써, 사용자가 편리하게 사용할 수 있는 장점이 있는 반면, 질의 전송 메시지 수가 증가하여 통신비용이 증가한다. 또한, 제한된 데이터 집계(aggregation) 질의만을 지원하여 효율적인 데이터 수집이 어려운 문제점이 존재한다. 이러한 문제점을 해결하기 위하여 SwissQM[4]이 개발되었다. SwissQM은 Java 가상 머신(Java Virtual Machine: JVM)[5] 기반 바이트코드(bytecode) 명령어와 인터프리터(interpreter)를 통하여 질의 수행에 필요한 메시지의 크기를 감소시키며, 데이터 집계 질의를 통하여 데이터 전송비용을 감소시킨다. 그러나 SwissQM은 다양한 센서 노드 플랫폼을 지원하지 못할 뿐만 아니라, 데이터 집계 질의 수행 시 불필요한 데이터 전송에 따른 데이터 집계 질의의 효율성이 저하되는 문제점이 존재한다. 따라서 본 논문에서는 이러한 문제점을 해결하기 위하여 에너지 효율성을 지원하는 바이트코드 기반 데이터 서비스 미들웨어(Bytecode based Data Service Middleware)를 개발한다. 제안하는 미들웨어는 첫째, 하드웨어 추상화를 지원하는 데이터 관리자와 다양한 하드웨어 플랫폼에서 동작할 수 있는 최소한의 바이트코드 명령어를 통하여, 센서 노드에 최적화된

기능만을 제공한다. 둘째, 불필요한 데이터를 제거함으로써 데이터 전송량을 최소화하는 데이터 집계 질의처리 및 데이터 필터링을 통하여, 센서 노드의 에너지 효율성을 향상시킨다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구인 TinyDB와 SwissQM을 살펴보고, 3장에서는 본 연구에서 제안한 바이트코드 기반 데이터 서비스 미들웨어의 설계 및 구현에 대하여 기술한다. 4장에서는 제안하는 미들웨어의 성능평가와 함께 개발한 미들웨어의 검증을 위한 침입 감지 응용 시스템을 구현한다. 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

지오센서 네트워크에 활용되는 센서 노드는 제한된 전력량과 같은 한정된 자원을 지닌다. 따라서 제한된 에너지를 효율적으로 사용하기 위한 지오센서 네트워크 미들웨어에 대한 연구가 활발히 진행 중이다.

2002년 Cornell 대학에서 연구한 데이터베이스 기반 미들웨어인 Cougar[6]는 지오센서 네트워크용 분산 데이터 처리 시스템으로써 SQL과 유사한 질의 언어를 지원한다. 아울러, 네트워크가 변경되었을 경우 동적으로 적응하는 유연성, 확장성을 제공하며, 장애가 발생하였을 경우 동적으로 장애를 처리하는 기능을 제공한다. 그러나 Cougar는 서버설치형(server-side) 미들웨어로써 센서 노드들이 수집한 정보를 여과 없이 서버로 전송하기 때문에, 서버 시스템의 비효율성을 초래하는 단점이 있다.

2005년 Washington 대학에서 개발한 모바일 에이전트 지원 미들웨어인 Agilla[7]는 인접 노드 정보를 유지함으로써 효율적인 무선 네트워크 환경을 지원하며, 동시에 노드 내의 여러 에이전트들을 통하여 다양한 응용 계층의 요청을 처리한다. 아울러, 이주(migration) 메커니즘을 통해, 새로운 코드를 센서 노드에게 배포하여 동적으로 미들웨어를 갱신한다. 그러나 Agilla는 미들웨어의 기능이 갱신되었을 경우, 미들웨어의 전체코드를 주변의 센서 노드로 전송하기 때문에 통신의 오버헤드를 초래하는 단점이 있다.

2005년 Berkeley 대학에서 개발한 데이터베이스 기반 미들웨어인 TinyDB는 센서 네트워크를 가상

데이터베이스로 간주하는 미들웨어로써, SQL과 유사한 질의 언어를 지원한다. 또한, TinyDB는 SRT(Semantic Routing Tree) 기법을 통해 질의 수행에 적합한 노드들에 한하여 질의 결과를 전송함으로써, 데이터 전송비용을 감소시킨다.[8] 그러나 TinyDB는 SQL과 유사한 고수준의 질의어를 사용함에 따라 소형 센서 노드에서 사용하기엔 비교적 큰 전송 메시지 수와 처리 속도를 지니고 있으며, 제한된 데이터 집계 기법만을 지원하여 효율적인 데이터 수집이 어려운 문제점이 존재한다. 또한 SRT를 사용함에 따라 다양한 종류의 질의를 동시에 처리하지 못하며, 센서 노드가 빈번히 이동할 경우 SRT의 유지비용이 급격히 증가하는 단점이 존재한다.

이러한 TinyDB의 문제점을 해결하기 위하여 제안된 SwissQM은 2007년 스위스 Zürich 공과대학에서 개발한 가상 머신 기반 미들웨어로써, Java 가상 머신 기반 바이트코드 명령어와 인터프리터를 통하여 질의 수행에 필요한 메시지의 크기를 감소시킨다. 또한, 최대 6개의 질의를 동시에 처리 가능하며, in-network 데이터 집계 질의를 통하여 센서 노드들간의 통신 메시지를 최소화한다. 그림 1은 SwissQM의 전체 구조를 나타낸다. SwissQM은 바이트코드 인터프리터(bytecode interpreter)와 데이터 저장 및 연산, 전송을 위한 스택(stack), 통신 버퍼(transmission buffer), 시놉시스(synopsis)로 구성된다. 바이트코드 인터프리터는 바이트코드 명령어를 실행하고, 각 명령어에 적합한 커맨드와 이벤트를 호출한다. 스택은 바이트코드 명령어의 연산자를 저장하며, 기본적인 데이터 연산을 수행하기 위한 메모리 구조이다. 통신 버퍼는 이웃 센서 노드와의 데이터 송수신을 위한 메모리 구조이며, 스택의 연산 결과 혹은 시놉시스의 데이터를 저장한다. 시놉시스는 합계, 평균과 같은 데이터 집계 연산을 수행하기 위한 메모리 구조로써, 각 그룹별 데이터 집계 연산 결과를 저장한다.

그러나 SwissQM은 특정 센서 노드 플랫폼에 의존적으로 개발되었기 때문에, 센서 네트워크의 주변 상황 및 하드웨어의 추가나 변경에 동적으로 대처하지 못한다. 또한, SwissQM은 데이터 집계 질의가 수행되는 동안 새로운 데이터가 수신될 경우, 새롭게 수신된 데이터에 대하여 데이터 집계 질의를 수행하지 못한다. 이는 수신된 새로운 데이터에 대

하여 데이터 집계 질의를 수행하지 않고 상위 노드로 그대로 전송하는 문제점을 가지고 있다.

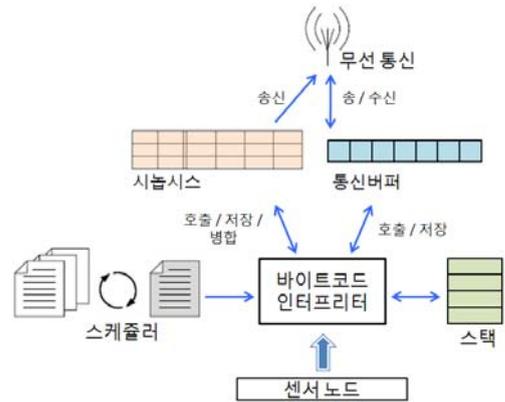


그림 1. SwissQM 전체 구조

3. 에너지 효율성을 지원하는 바이트코드 기반 데이터 서비스 미들웨어

효율적인 센서 네트워크 미들웨어를 개발하기 위해서는 다음과 같은 사항이 고려되어야 한다. 첫째, 센서 네트워크 시스템의 확장성을 지원하기 위해, 센서 노드 하드웨어와 응용 시스템에 의존적이지 않은 유연한 시스템을 제공해야 한다. 기존 센서 네트워크 미들웨어는 센서 노드 플랫폼에 따라 미들웨어의 사용 가능성이 결정되기 때문에, 센서 네트워크의 하드웨어의 추가나 변경에 동적으로 대처하지 못한다. 따라서 이기종 센서 사용을 지원하기 위한 하드웨어 추상화와 다양한 하드웨어 플랫폼에서 동작할 수 있는 명령어 집합(instruction set)이 필수적이다. 둘째, 센서 노드는 배터리와 같은 제한된 에너지 자원을 사용하기 때문에, 에너지 효율성 향상이 필수적이다. 따라서 대표값 선정을 통하여 중복된 데이터를 제거하는 것뿐만 아니라, 사용자가 원하지 않는 불필요한 데이터를 제거하여 데이터 전송량을 최소화하는 데이터 집계 질의에 대한 추가적인 연구가 필요하다.

이에 따라 본 연구에서는 센서 네트워크에서 에너지 효율성을 지원하는 바이트코드 기반 데이터 서비스 미들웨어를 개발한다. 개발하는 미들웨어는 첫째, 센서 노드에 최적화된 기능을 제공하기 위해, 하드웨어 추상화를 지원하는 데이터 관리자와 최소한의 Java 가상 머신 기반 바이트코드 명령어를 지

니다. 이를 통하여 다양한 센서 노드 플랫폼에 대한 추상화를 지원하며, 최소한의 명령어 집합을 통하여 센서 노드의 자원 소모량을 감소시킵니다. 둘째, 센서 노드의 에너지 효율성 향상을 위해, 데이터 집계 질의처리 및 데이터 필터링(filtering) 기능을 제공하는 질의처리 관리자 및 데이터 필터링 관리자를 지원한다. 이를 통하여 불필요한 데이터를 제거하고 사용자가 원하는 데이터만을 전송함으로써 데이터 전송량을 감소시킵니다.

제안하는 데이터 서비스 미들웨어는 대표적인 센서 네트워크 운영체제인 TinyOS[9]를 기반으로 동작되며, 각 센서 노드에 탑재되어 게이트웨이(gateway)와 센서 노드와의 유연한 연동과 함께 센서 노드의 에너지 효율성을 지원한다. 그림 2는 제안하는 데이터 서비스 미들웨어의 전체적인 구조를 나타낸다. 데이터 서비스 미들웨어는 데이터 관리자, 바이트코드 관리자, 질의처리 관리자, 데이터 필터링 관리자 및 통신 관리자를 포함하며, 수집된 센서 데이터로부터 최소한의 명령어 집합으로 집계 처리를 수행할 수 있는 경량화된 센서 네트워크 미들웨어이다. 한편, 게이트웨이는 사용자가 작성한 질의를 컴파일하여 센서 노드로 전송하며, 센서 노드로부터 전송된 질의 결과를 처리하여 사용자가 확인할 수 있도록 GUI 인터페이스를 지원하는 클라이언트(client) 역할을 수행한다. 본 연구에서는 Apache ANT를 이용하여 게이트웨이의 GUI를 제공한다. Apache ANT는 Apache사에서 제작한 빌더툴(builder tool)로서, Java 컴파일 및 데이터베이스에 대한 손쉬운 접근 기법을 제공한다.

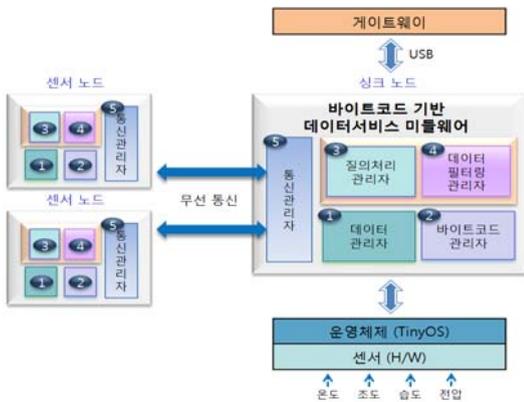


그림 2. 에너지 효율성 지원 바이트코드 기반 데이터 서비스 미들웨어 전체 구조

3.1 데이터 관리자

기존 데이터 서비스 미들웨어는 특정 센서 노드 플랫폼에 의존적이다. 이로 인해, 센서 노드 플랫폼에 따라 미들웨어 사용 가능성이 결정되는 문제점이 발생한다. 또한, 기존 센서 네트워크 미들웨어는 그림 3과 같이 수집되는 데이터를 별도의 데이터 관리자를 통하여 관리하지 않음으로 인하여 메모리 공간의 비효율성을 초래한다. 이는 한정된 자원을 가지고 있는 센서 노드에는 치명적인 문제이다. 이를 해결하기 위해 다양한 하드웨어 플랫폼의 추상화를 지원하는 데이터 수집 컴포넌트와 효율적인 메모리 관리를 위한 데이터 관리 컴포넌트를 설계한다.

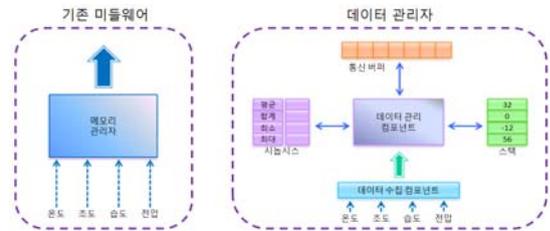


그림 3. 데이터 관리자 전체 구조

먼저, 데이터 수집 컴포넌트는 각 센서 노드에 최적화된 데이터 수집 기능을 모듈화(modulization)하여 구성한 후, 프로그램 컴파일 시, 각 플랫폼에 적합한 모듈만을 연결(wiring)함으로써 다양한 하드웨어 플랫폼에 대한 추상화를 지원하는 컴포넌트이다. 현재 개발한 데이터 수집 컴포넌트에서 지원하는 각 센서 플랫폼별 수집 가능 항목은 표 1과 같다.

표 1. 센서 플랫폼별 수집 가능 항목

수집 항목	센서 플랫폼			
	TelosB	Mica2/MicaZ		
		MTS300	MTS400	MTS420
조도	○	○	○	○
온도	○	○	○	○
습도	○	N/A	○	○
전압	○	○	○	○
음성	N/A	○	N/A	N/A
기압	N/A	N/A	△	△
가속도	N/A	N/A	△	△
GPS	N/A	N/A	N/A	△
기타	Knote-Door: 도어 개폐, Knote-PIR: 적외선			

○ : 지원, △ : 추가 확장 지원, N/A : H/W 미지원

아울러, 데이터 관리 컴포넌트는 데이터 수집 컴포넌트로부터 수집된 데이터를 관리하며, 효율적인 메모리의 사용을 위하여 수집된 데이터를 스택, 통신 버퍼, 시놉시스로 나누어 관리한다. 그림 4는 데이터 관리 컴포넌트의 전체 구조를 나타낸다. 첫째, 스택은 센서 노드로부터 수집된 센싱값 등 피연산자를 저장 및 관리하기 위한 메모리 구조로써, 기본적인 데이터 연산을 수행한다. 둘째, 통신 버퍼는 “매 2초마다 조도를 측정하라”와 같은 데이터 수집 명령어로 수집된 데이터들의 전송을 위한 메모리 구조이다. 마지막으로, 시놉시스는 “수집된 조도의 평균을 구하라”와 같은 데이터 집계 질의를 수행하기 위한 메모리 구조로써, 각 그룹별 데이터 집계 연산 결과를 저장한다. 시놉시스에서 지원하는 데이터 집계 연산은 수(Count), 최대(Max), 최소(Min), 합계(Sum), 평균(Avg), 분산(Var)이며, 최대 4개의 집계 그룹에 대한 데이터 집계 질의 결과를 유지할 수 있다.

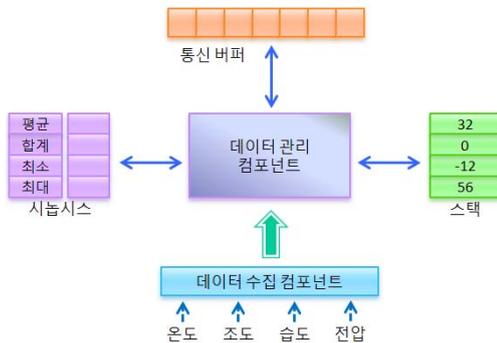


그림 4. 데이터 관리 컴포넌트 구조

3.2 바이트코드 관리자

센서 노드의 에너지는 미들웨어의 함수를 호출하거나, 센서 노드간의 데이터 통신을 수행할 때, 또는 SQL과 같은 질의 언어를 컴파일 할 때 크게 소비된다. 따라서 센서 노드들 간의 메시지 통신비용과 컴파일에 필요한 자원 소모량을 감소시키기 위한 명령어 집합이 필수적이다. 또한 센서 네트워크의 확장성을 위해 특정 센서 노드 플랫폼에 의존적이지 않은 유연한 시스템을 제공해야 한다. 이를 위하여 본 연구에서는 최소한의 Java 가상 머신 기반 바이트코드 명령어 집합을 이용함으로써 센서 노드의 에너지 효율성과 센서 네트워크의 확장성을 지원한다.

다. 제안하는 바이트코드 관리자는 기본적인 연산을 수행하기 위한 기본 바이트코드 명령어 집합과 센서 네트워크의 에너지 효율성 향상을 위한 데이터 집계 바이트코드 명령어 집합으로 구성된다. 표 2는 Java 가상 머신 명령어와 본 연구에서 설계한 바이트코드 명령어 집합에 대한 비교를 나타낸다.

표 2. 명령어 집합 비교

Java 가상 머신 명령어 집합	명령어수	바이트코드 명령어 집합	명령어수	
Operand Stack Management	9	기본 명령어	Stack	4
Arithmetic	42		Arithmetic	3
Control Transfer	23		Control	4
Load and Store	32		Buffer	6
Type Conversion	9		Sensor	7
Object Creation	27	데이터 집계	Aggregation	1
			Filtering	2
			Top-K	5
Method Invocation and Return	4			
Total	142	Total	32	

먼저, 기본 바이트코드 명령어 집합은 기본적인 연산을 수행하기 위하여 Java 가상 머신 명령어를 기반으로 생성한 최소한의 명령어 집합이다. 기본 바이트코드 명령어 집합은 표 3에서와 같이 Stack, Arithmetic, Control, Buffer, Sensor 명령어 집합으로 구성되어 있다. 표에서 x, y, z는 스택에 저장되어 있는 상위 엘리먼트(element)의 값을 의미하며, a는 x, y, z를 제외한 기존 스택에 저장되어 있는 element를 의미한다. Stack 명령어 집합은 스택 데이터의 복사, 제거, 삽입을 수행한다. Arithmetic 명령어 집합은 스택 데이터의 간단한 산술 연산을 수행한다. Control 명령어 집합은 각 질의 명령을 제어하는 역할을 수행한다. Buffer 명령어 집합은 버퍼, 개요의 데이터를 관리하는 역할을 수행한다. 마지막으로 Sensor 명령어 집합은 센서에서 측정된 데이터를 스택에 저장한다.

한편, 데이터 집계 바이트코드 명령어 집합은 센서 네트워크의 에너지 효율성 향상을 위하여 본 연구에서 추가적으로 설계한 명령어 집합이다. 데이터 집계 바이트코드 명령어 집합은 표 4에서와 같이 Aggregation, Filtering, Top-K 명령어 집합으로 구성되어 있다. Aggregation 명령어 집합은 데이터

표 3. 기본 바이트코드 명령어 집합

명령어 집합	분 류	스택 상태
Stack	dup	$a, x \Rightarrow a, x, x$
	pop	$a, x \Rightarrow a$
	swap	$a, y, x \Rightarrow a, x, y$
	ipushw i	$a \Rightarrow a, i$
Arithmetic	iadd	$a, y, x \Rightarrow a, y + x$
	iinc	$a, x \Rightarrow a, x + 1$
	iand	$a, y, x \Rightarrow a, y \& x$
Control	if_icmpeq i	$a, y, x \Rightarrow a, \text{jump if } y = x$
	if_icmplt i	$a, y, x \Rightarrow a, \text{jump if } y < x$
	ifgt	$a, x \Rightarrow a, \text{jump if } x > 0$
	goto	$a \Rightarrow a, \text{jump always}$
Buffer	iload i	$a \Rightarrow a, \text{buf}[i]$
	istore i	$a, x \Rightarrow a \text{ and } \text{buf}[i] = x$
	iload_sy i	$a \Rightarrow a, \text{synopsis}[i]$
	istore_sy i	$a, x \Rightarrow a \text{ and } \text{synopsis}[i] = x$
	send_tb	send transmission buffer
	send_sy	send synopsis
Sensor	get_nodeid	$a \Rightarrow a, \text{nodeid}$
	get_parent	$a \Rightarrow a, \text{parent}$
	get_light	$a \Rightarrow a, \text{light}$
	get_temp	$a \Rightarrow a, \text{temp}$
	get_humid	$a \Rightarrow a, \text{humid}$
	get_noise	$a \Rightarrow a, \text{noise}$
	get_voltage	$a \Rightarrow a, \text{battery voltage}$

표 4. 데이터 집계 바이트코드 명령어 집합

명령어 집합	분 류	스택 상태
Aggregation	merge	$a, \text{aggm}, \dots, \text{agg1}, m, n \Rightarrow a$ n: number of groups m: number of aggregates aggi: code of aggregate
Filtering	m_filter i i	$a \Rightarrow a \text{ and } \text{min} = i \text{ max} = i$
	a_filter	$a \Rightarrow a \text{ and } \text{min} = \text{auto} \text{ max} = \text{auto}$
Top-K	get_light_topk i	$a \Rightarrow a, \text{light_top1}, \dots, \text{light_topk}$
	get_temp_topk i	$a \Rightarrow a, \text{temp_top1}, \dots, \text{temp_topk}$
	get_noise_topk i	$a \Rightarrow a, \text{noise_top1}, \dots, \text{noise_topk}$
	get_tone_topk i	$a \Rightarrow a, \text{tone_top1}, \dots, \text{tone_topk}$
	get_voltage_topk i	$a \Rightarrow a, \text{voltage_top1}, \dots, \text{voltage_topk}$

집계처리 연산을 수행하며, Filtering 명령어 집합은 불필요한 데이터를 제거하는 역할을 수행한다. 마지막으로 Top-K 명령어 집합은 k개의 최상위 데이터를 추출하는 Top-K 질의를 수행한다.

3.3 질의처리 관리자

질의처리 관리자는 앞서 설계한 바이트코드 명령어를 수행하기 위한 관리자로서, 일반 질의처리 컴포넌트와 데이터 집계 질의처리 컴포넌트로 구성된다. 일반 질의처리 컴포넌트는 앞서 설계한 기본 바이트코드 명령어 집합을 수행하며, 데이터 집계 질의처리 컴포넌트는 데이터 집계 바이트코드 명령어 집합을 수행한다. 일반 질의처리 컴포넌트는 기본 바이트코드 명령어 집합인 Stack, Arithmetic, Control, Buffer, Sensor 명령어 집합에 대한 질의를 수행한다. 예를 들면, 그림 5와 같이 “50초의 주기로 측정하여, 온도가 20도 이하인 경우 해당 센서 노드의 ID와 온도를 전송하라.”와 같은 사용자 질의를 수행한다. 그림과 같이 바이트코드 명령어를 이용하여 질의가 요청되면, 일반 질의처리 컴포넌트는 다음과 같은 수행과정으로 처리한다. 과정 ①, ②는 해당 센서 노드의 ID를 스택에 넣고, 스택의 값을 통신 버퍼 0의 인덱스 위치로 이동한다. 과정 ③, ④는 온도를 측정하여 스택에 넣고, 스택의 top 위치의 값을 복사한다. 과정 ⑤는 상수 20값을 스택에 넣는다. 과정 ⑥은 스택에서 두 개의 값을 꺼내어 비교하고, top 위치의 값보다 top-1 위치의 값이 크면 end 부분을 수행한다. 과정 ⑦은 스택에서 1개의 값을 통신 버퍼 1의 인덱스 위치로 이동한다. 이와 같이, 일반 질의처리 컴포넌트는 데이터의 단순 처리를 지원하며, 알고리즘의 종류는 바이트코드 관리자에서 언급하였다.

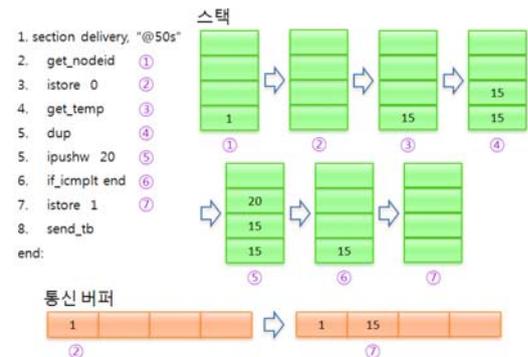


그림 5. 일반 질의처리 수행과정

데이터 집계 질의처리 컴포넌트는 앞서 설계한 Aggregation 명령어 집합을 수행하기 위한 집계 질의처리 모듈과 Top-K 명령어 집합에 대한 질의를 수행하기 위한 Top-K 질의처리 모듈로 구성되어 있다. 첫째, 집계 질의처리 모듈은 센서 노드에서 수집한 데이터 집합의 특징이나 경향을 나타내는 대표값을 선정한다. 표 5는 데이터 집계 질의처리에서 지원하는 대표값의 종류를 나타낸다.

집계 질의처리 모듈은 하위 노드에서 받은 데이터와 자신의 수집한 데이터를 하나의 값으로 집계한다. 예를 들면, 그림 6과 같이 “30초의 주기로 온도의 평균값을 전송하라.”와 같은 사용자 질의를 수행한다. 그림과 같이 바이트코드를 이용하여 질의가 요청되면, 집계 질의처리 모듈은 다음과 같은 수행과정으로 처리한다. 과정 ①, ②는 온도를 측정하여 스택에 넣고, 그 값을 꺼내어 자신의 시놉시스로 이동시킨다. 과정 ③, ④, ⑤는 집계 코드(5 : 평균을

의미), 집계할 데이터의 개수, 집계 그룹 번호(온도 그룹화를 의미)를 설정한다. 과정 ⑥은 스택에 설정된 값을 이용하여 자신의 시놉시스 데이터와 하위 노드로부터 전송받은 데이터를 집계하여 시놉시스의 데이터를 갱신한다.

집계 질의처리 모듈은 데이터 집계 질의처리 수행 시, 시놉시스에 저장된 집계 질의 결과가 최대한도인 4개를 초과할 경우, 현재까지 저장된 시놉시스를 부모 노드로 전송하여 데이터의 손실을 방지한다. 그림 7은 부모 노드의 ID를 집계 그룹으로 설정하여 동일한 집계 그룹에서 조도의 최대값을 추출하는 데이터 집계 질의의 예시를 나타낸다. 그림 7(a)는 시놉시스 저장 공간 초과 현상이 발생하는 센서 네트워크의 구조와 각 센서 노드에서 측정된 조도 값을 나타낸다. 그림 7(b)는 시놉시스 저장 공간 초과 현상이 발생하였을 때 기존에 저장되어 있는 시놉시스를 부모 노드로 전송하고 다시 데이터 집계 질의를 수행하는 것을 나타낸다. 그림 7(b)에서 1번 노드는 2번 노드로부터 3개의 질의 결과가 저장된 시놉시스를 전송받고, 3번 노드로부터는 2개의 질의 결과가 저장된 시놉시스를 전송받는다. 이에 따라 1번 노드는 시놉시스 저장 공간 초과 현상이 발생된다. 따라서 1번 노드는 우선적으로 전송받은 2번 노드로부터의 시놉시스를 부모 노드로 전송하여 시놉시스를 초기화한다. 이후 1번 노드는 3번 노드에서 전송된 시놉시스와 자신의 데이터에 대하여 다시 데이터 집계 질의를 수행한다.

둘째, Top-K 질의처리 모듈은 센서 데이터 중에서 가장 높은 k개의 값을 찾는 Top-K 질의[10]를 수행한다. Top-K 질의는 여러 모니터링 분야에서 제한된 에너지를 효율적으로 사용하기 위한 알고리즘으로 유용하게 사용된다. 예를 들면, “오염된 지역에서의 오염 제거를 위해 가장 많은 오염이 발생한 k개의 지역을 찾아라.”와 같이 사용될 수 있다. 그림 8은 Top-K 질의를 센서 노드 상에서 수행하는 과정을 나타낸다. 메모리상에서 하위노드로부터 받은 데이터를 2k개의 메모리를 이용하여 정렬한다. 이 후, 메모리에 있는 정렬 데이터를 스택에 저장한다. 스택에 저장된 k번째의 데이터와 자신이 수집한 데이터를 비교하여 스택의 k번째 데이터가 크다면 스택에 있는 데이터를 상위 노드에 전송한다. 자신의 데이터가 크다면 k번째 데이터를 제외한 k-1개의 데이터와 자신의 데이터를 전송한다.

표 5. 집계 질의처리 대표값 종류

대표값	의미	계산식
데이터 수	전체 데이터 수	count = N
총합	전체 데이터 합	$sum = \sum_{i=0}^N Data_i$
평균	전체 데이터의 평균	$avg = \frac{1}{N} \sum_{i=0}^N Data_i$
분산	데이터와 전체 데이터 평균의 차이의 합	$var = \frac{1}{N} \sum_{i=0}^N (Data_i - Avg)^2$
최소값	전체 데이터 중 최소값	$min = Min(Data_1, Data_2, \dots, Data_N)$
최대값	전체 데이터 중 최대값	$max = Max(Data_1, Data_2, \dots, Data_N)$

* N = 전체 데이터 수

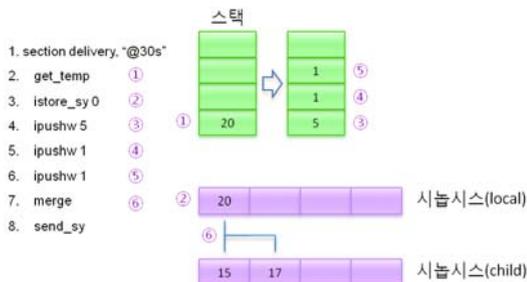


그림 6. 데이터 집계 질의처리 수행과정

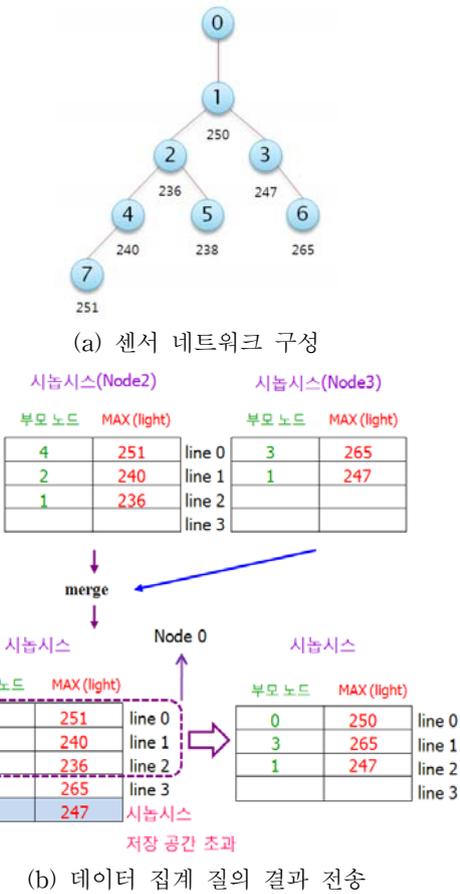


그림 7. 데이터 집계 질의처리 예시

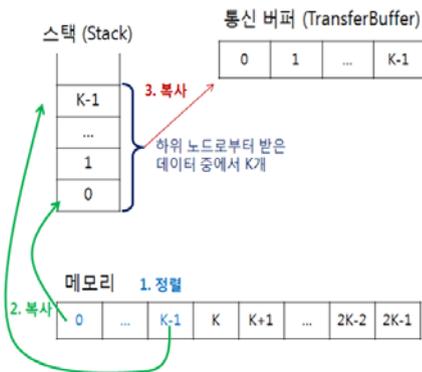


그림 8. Top-K 질의처리 수행과정

3.4 데이터 필터링 관리자

사용자는 일반적인 상황에서 수집한 데이터보다 특수한 상황에서 수집된 데이터에 많은 관심을 나

타낸다. 예를 들어 화재감시 시스템의 경우, 건물 내에서 일반적으로 측정되는 평균적인 온도 데이터보다, 화재 발생에 의하여 급격히 상승하는 온도 데이터에 많은 관심을 나타낸다. 즉, 평균 온도의 경우, 이전까지 수집된 데이터 집합의 대표 값을 측정함으로써 사용자가 예측 가능하지만, 화재 발생 시 온도 데이터는 대표 값에 의해 사용자가 예측하지 못한다. 따라서 사용자가 원하지 않는 데이터란 특정 범위 내에 존재하는 데이터, 또는 이전에 측정된 데이터 대표 값에 의해 표현 가능한 데이터를 의미한다. 따라서 본 연구에서는 불필요한 데이터를 제거함으로써 데이터 전송비용을 감소시킬 수 있는 데이터 필터링 관리자를 설계한다. 제안하는 데이터 필터링 관리자는 사용자가 원하지 않는 영역 내의 데이터를 제거하는 범위 필터링 컴포넌트와 이전 데이터와의 오차 범위를 고려하여 데이터의 전송 여부를 결정하는 대표값 필터링 컴포넌트로 구성된다.

첫째, 범위 필터링 컴포넌트는 사용자가 일정 범위의 데이터를 제거하고자 할 때 사용되는 컴포넌트이다. 이를 위해 사용자는 데이터 범위를 표현하기 위한 데이터의 최소값 및 최대값을 지정한다. 센서 노드는 수집된 데이터가 사용자 지정 범위 내에 포함되는지 확인하고, 만약 해당 범위 이외의 데이터가 수집되면, 이를 반환한다. 예를 들어 사용자 범위가 10~20일 때, 수집된 데이터가 14인 경우는 데이터를 반환하지 않고, 수집된 데이터가 21인 경우 21을 반환한다.

둘째, 대표값 필터링 컴포넌트는 사용자가 예측 가능한 데이터를 제거하고자 할 때 사용되는 컴포넌트이다. 즉, 이전 데이터와의 오차 범위를 고려하여 데이터의 전송 여부를 결정함으로써 데이터 전송 횟수를 감소시킨다. 예를 들어, 실제 온도가 10.5°C라고 할 때, 동일 기종의 센서 A와 B를 통해 온도를 측정한다고 가정하자. 이때, 센서 노드의 물리적 특성상 오차가 발생하기 때문에, A와 B는 동일 기종임에도 각각 10.4°C, 10.6°C를 온도 값으로 측정할 수 있다. 그러나 일반적으로 10.4°C와 10.6°C는 크게 차이가 없으며, 사용자는 센서의 물리적 특성을 고려하고 있기 때문에, 실제 온도가 10.4°C일 때 온도 데이터 값이 10.6°C라고 해서 크게 문제가 되지는 않는다. 이와 같은 특징을 이용하여 대표값 필터링은 센서가 새롭게 측정한 데이터

값과 이전까지의 데이터를 기반으로 예측한 데이터 값과의 차이가 미리 정한 에러 범위 안에 있을 경우, 새로 측정된 데이터를 전송하지 않는다.

한편, 대표값 필터링은 정확한 데이터 예측값 선정을 위하여 대표적인 선형 필터인 칼만(Kalman) 필터[11]를 사용한다. 칼만 필터는 높은 예측 정확도 때문에, 데이터 필터링이 필요한 다양한 응용 분야에서 활용되고 있다. 칼만 필터의 데이터 예측은 수식 1에 의해 계산된다. 여기서, avg_update 는 현재 측정값을 이용하여 예측한 데이터의 평균값, avg_prev 는 이전에 예측한 데이터 평균값을 의미하며, sd_update 는 현재 측정된 표준편차, sd_prev 는 이전에 예측한 데이터의 표준편차를 의미한다. 또한 SensingError는 실제 센서에서 데이터 측정 시 발생하는 에러를 의미하며, Data는 현재 감지한 센싱값을 의미한다. 예를 들어, $avg_prev = 40$, $sd_prev = 1$, $Data = 41$, $SensingError = 0.5$, 데이터 필터링 한계 오차율 = 3% 라고 가정하면, 수식 1에 의해 $avg_update = 40.8$, $sd_update = 5$ 가 된다. 따라서 예측값 40.8과 현재 측정값 41 사이의 예측 오차는 $(41-40.8)/40.8 = 0.0049$, 즉 약 0.5% 정도의 오차를 나타낸다. 따라서 오차율이 데이터 필터링 한계 오차율 3%보다 작기 때문에, 현재 데이터는 전송하지 않는다.

$$avg_update = \frac{avg_prev * SensingError^2 + Data * (sd_prev)^2}{(sd_prev)^2 + SensingError^2}$$

$$sd_update = \frac{(sd_prev)^2 + SensingError^2}{(sd_prev)^2 * SensingError^2}$$

수식 1. 대표값 필터링의 데이터 예측값 선정

3.5 통신 관리자

통신 관리자에서는 기본적으로 TinyOS에서 제공하는 기본 통신 프로토콜인 B-MAC을 이용하여 데이터를 송수신한다. 아울러 TinyOS에서 제공하는 대표적인 멀티홉 라우팅 프로토콜인 Mint Route[12]를 이용하여 라우팅을 구성한다.

한편, SwissQM은 TinyOS에서 발생하는 인터셉트 이벤트(intercept event)를 통하여 데이터 집계 질의를 수행한다. 인터셉트 이벤트는 TinyOS에서 데이터를 수신할 경우 발생하는 이벤트이며, 이를 통해서만 수신 데이터에 대한 처리가 가능하다. 이

는 TinyOS가 이벤트가 발생되었을 경우 동작되는 이벤트 구동식(event-driven) 방식이기 때문이다. 이에 따라 SwissQM은 데이터 집계 질의가 수행되는 동안 새로운 데이터가 수신되어 인터셉트 이벤트가 발생될 경우, 새롭게 수신된 데이터에 대하여 데이터 집계 질의를 수행하지 못한다. 이는 동일한 시간대에서의 데이터 수신 충돌은 MAC의 기본 기능인 충돌 회피(collision avoidance)를 통하여 해결 가능하지만, 데이터의 일관성(consistency) 측면에서 데이터 집계 질의가 실행되는 동안 새로운 데이터 집계 질의를 수행할 수 없기 때문이다. 따라서 SwissQM은 데이터 집계 질의 수행 시 수신된 새로운 데이터에 대하여 데이터 집계 질의를 수행하지 않고, 상위 노드로 그대로 전송하는 문제점을 가지고 있다. 그림 9는 이러한 기존 데이터 집계 질의의 문제점을 나타낸다.

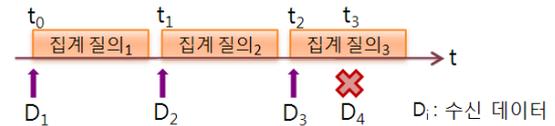


그림 9. 기존 데이터 집계 질의의 문제점

이를 해결하기 위하여, 통신 관리자는 집계 동기화 버퍼(SyncResultBuffer)를 이용하여 미집계 데이터에 대한 데이터 집계 질의를 수행한다. 그림 10은 부모 노드의 ID를 집계 그룹으로 설정하여 조도의 최대값을 추출하는 데이터 집계 질의를 수행 시, 통신 관리자를 이용하여 미집계 데이터에 대한 데이터 집계 질의를 수행하는 것을 나타낸다.

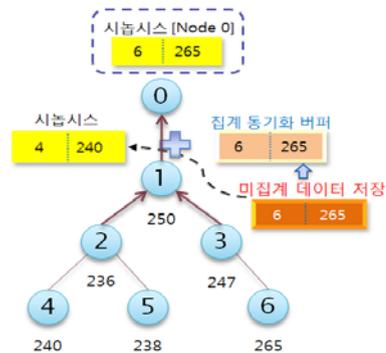


그림 10. 통신 관리자의 미집계 데이터 처리 예시

4. 검증 및 성능평가

본 절에서는 제안하는 에너지 효율성 지원 데이터 서비스 미들웨어에 대한 성능평가 및 침입 감지 응용 시스템 프로토타입의 구현을 통하여 개발한 미들웨어에 대한 검증을 수행한다. 성능 평가 대상은 기존의 가상 머신 기반 미들웨어인 SwissQM이며, 성능 평가 항목은 데이터 집계 질의 수행 시 센서 네트워크 크기에 따른 데이터 전송량, 평균 에너지 소모량이다. 아울러, 개발한 데이터 필터링 관리자를 적용한 데이터 전송 빈도에 대하여 성능 평가를 수행한다.(한편, 제안하는 에너지 효율성 지원 데이터 서비스 미들웨어는 BDSM(Bytecode based Data Service Middleware)으로 명명한다.) 성능 평가의 실험 환경은 표 6과 같으며, TinyOS에서 제공하는 무선 센서 네트워크 전용 시뮬레이터인 TOSSIM 시뮬레이터[13]에서 nesC[14] 프로그래밍 언어를 이용하여 구현 및 성능평가를 수행하였다. 성능평가는 100*100m의 정방향의 네트워크 영역에서, 센서 노드의 통신 범위는 대표적인 센서 노드인 MicaZ의 실내 평균 통신 거리인 25m로 수행하였고, 부모 노드의 ID를 집계 그룹으로 설정하여 최대 조도값을 측정하는 데이터 집계 질의를 수행하였다. 또한, 전체 센서 노드의 배치는 노드를 전체적으로 임의로 배치하는 임의 배치(random) 유형을 사용하였으며, TinyOS에서 제공하는 Mint Route를 이용하여 라우팅을 구성하였다.

표 6. 구현 환경

CPU	Intel Core2 Duo CPU 2.40GHz
Ram	2G
Simulator	TOSSIM
Compiler	gcc compiler

첫째, 네트워크 크기에 따른 데이터 전송량 측정을 위하여, 전체 노드 수를 32, 64, 128로 증가하면서 총 50주기까지 데이터 집계 질의처리를 위한 각각의 평균 메시지수를 측정하였다. 그림 11은 센서 네트워크의 크기 변화에 따른 데이터 집계 질의처리 메시지 수를 비교한 것이다. 이를 통하여 BDSM은 SwissQM보다 데이터 집계 질의처리 메시지 수가 최대 16% 감소함을 알 수 있다. 특히 BDSM의

경우, 데이터 집계 질의 수행 실패에 따른 불필요한 데이터 전송을 최소화함으로써, 전체 노드 수가 증가할수록 더 좋은 성능을 보인다. 결과적으로, 제안하는 BDSM은 효율적인 데이터 집계 질의처리를 통하여, 센서 네트워크 크기에 따른 데이터 전송량 성능평가 시 SwissQM보다 우수한 성능을 나타낸다.

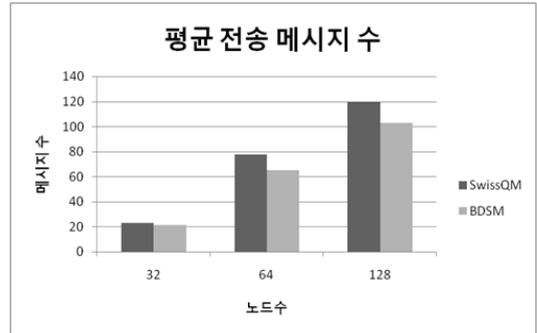


그림 11. 네트워크 크기에 따른 데이터 전송량

둘째, 네트워크 크기에 따른 평균 에너지 소모량 측정을 위하여, 전체 노드 수를 32, 64, 128로 증가하면서 총 50주기까지 데이터 집계 질의처리 수행 시 각 노드별 평균 에너지 소모량을 측정하였다. 성능 평가 도구로는 TOSSIM에서 제공되는 Power Profiling을 사용하였다. Power Profiling에서 에너지 소모량은 수식 2에 의해 계산된다. 여기서 currentradio[id]는 해당 센서 노드의 현재 radio 통신 에너지를 의미하며, Voltage와 Duration은 각각 동작 전압과 주기를 의미한다. 또한, CPUFREQ는 CPU의 동작 빈도를 의미한다.

$$e_{radio}[id] = \frac{current_{radio}[id] * Voltage * Duration}{CPU_{FREQ}}$$

수식 2. 에너지 소모량 측정 방법

그림 12는 센서 네트워크의 크기 변화에 따른 노드별 평균 에너지 소모량을 비교한 것이다. 특히, SwissQM은 센서 네트워크의 규모가 커질수록, 빈번한 데이터 전송에 의하여 에너지 소모량이 급격히 증가함을 알 수 있다. 반면, BDSM은 불필요한 데이터 전송을 최소화함으로써 SwissQM보다 뛰어난 에너지 효율성을 보인다. 결과적으로, 제안하는 BDSM은 효율적인 데이터 집계 질의처리 기법을

통하여, 센서 네트워크 크기에 따른 평균 에너지 소모량 성능평가 시 SwissQM보다 우수한 성능을 나타낸다.

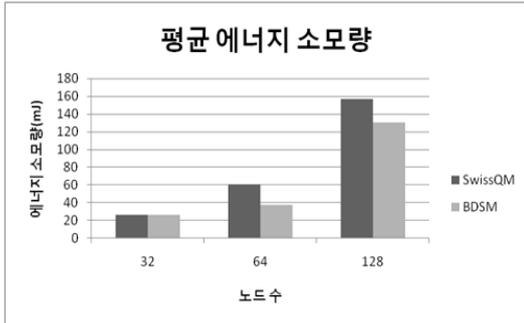


그림 12. 네트워크 크기에 따른 노드별 평균 에너지 소모량

셋째, 제안하는 BDSM에서 지원하는 데이터 필터링 기법을 적용한 데이터 전송량에 대한 성능평가를 수행한다. 범위 필터링의 경우 지정한 범위 내의 데이터가 모두 필터링 되므로 대표값 필터링에 대해서만 성능 평가를 수행하였다. 대표값 필터링 측정을 위하여 센서 노드의 수를 10, 20, 30으로 증가시키면서, 총 200주기 동안 성능평가를 수행하였다. 아울러, 성능평가를 위해 이전 데이터를 기준으로 주어진 변화율 범위 내에서 증가 혹은 감소하는 선형적 데이터 생성기를 구현하여 사용하였다. 예를 들어, 첫 번째 데이터가 10, 데이터 변화율이 10%의 경우, 두 번째 데이터는 9~11 사이의 범위에서 생성된다. 만약 두 번째 생성된 데이터가 9일 경우, 세 번째 데이터는 8.1~9.9 사이에서 생성된다. 이는 실제 센서 노드에서 감지되는 데이터와 유사한 데이터 변화율을 나타내기 위함이다. 생성되는 전체 데이터 수는 노드수 * 실험주기이며, 상위 노드에서의 라우팅을 위한 메시지 수를 고려할 때, 필터링 미적용시 전체 데이터 수는 전체 생성된 데이터 수 * 트리 깊이이다. 그림 13은 데이터 변화율이 4%이고 트리 깊이가 3일 때, 대표값 필터링을 적용한 데이터 전송량 측정 결과를 나타낸다. 대표값 필터링을 지원하지 않는 SwissQM의 경우, 노드 수에 따라 각각 6000, 12000, 18000번의 데이터 전송이 필요하다. 반면, 대표값 필터링을 적용한 BDSM의 경우, 2262, 2371, 2665 회의 평균 데이터 전송만이 발생한다. 따라서 제안하는 BDSM은 대표값 필터링

을 통하여 기존 SwissQM보다 데이터 전송량을 감소시키고, 노드 수가 증가하더라도 일정한 데이터 전송률을 나타낼 수 있다.

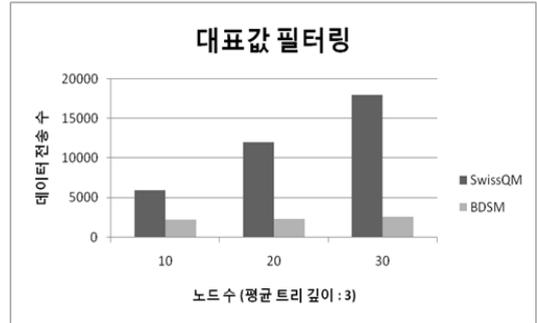
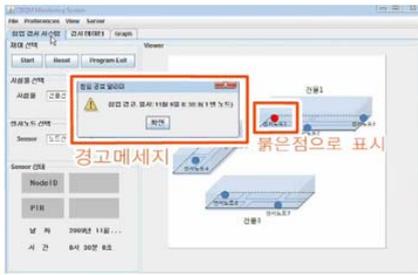
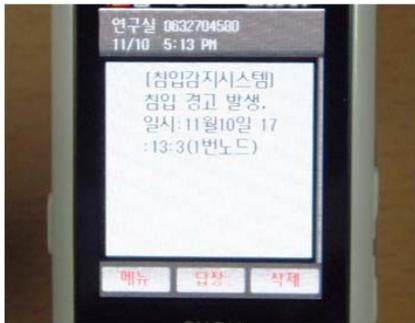


그림 13. 대표값 필터링을 적용한 데이터 전송량

넷째, 개발한 미들웨어의 검증은 위하여 BDSM을 토대로 한 침입 감지 응용 시스템 프로토타입을 구현하였다. 구현한 응용 시스템 프로토타입은 게이트웨이의 GUI 인터페이스와 센서 노드 미들웨어로 구성된다. 게이트웨이의 GUI 인터페이스에서는 사용자로부터 질의를 기계어로 번역하여 센서 노드로 전송한다. 센서 노드 미들웨어는 침입 여부를 판단하여 데이터를 유효한 형태로 가공하여 게이트웨이로 전송한다. 또한 침입 감지를 위하여 Knote-Door 센서와 Knote-PIR 센서를 사용하였다. Knote-Door 센서는 도어 개폐 유무를 확인하며, Knote-PIR 센서는 적외선 감지를 통하여 사물의 움직임을 감지한다. 화재 감시 응용 시스템 프로토타입의 인터페이스는 그림 14(a)와 같다. 크게 침입 감시 시스템 화면, 감시데이터 화면으로 구성되고, 위험 상황 경보를 위한 인터페이스가 존재한다. 아울러, 전체 센서 노드의 감시 상황을 확인할 수 있는 기능을 제공한다. 이는 현재 센서 노드가 배치된 건물이나 방의 조감도를 바탕으로 하여 사용자가 쉽게 이해할 수 있는 직관적인 인터페이스를 제공한다. 침입 상황 발생시, 그림과 같이 경고 창을 띄워 해당 센서 노드의 식별번호를 나타내고, 동시에 뷰어 창에 침입이 발생한 센서 노드의 위치를 붉은색 원으로 표시함으로써 사용자에게 경고 메시지를 전달한다. 아울러, 그림 14(b)와 같이 문자 메시지(SMS)를 미리 등록된 사용자 휴대 전화로 전송하여 신속한 대처가 가능하도록 한다.



(a) 감시 시스템 인터페이스



(b) 침입 경고 SMS 서비스

그림 14. 침입 감지 응용시스템 프로토타입 인터페이스

마지막으로, 개발한 BDSM 및 기존 미들웨어의 특징을 표 7과 같이 비교 분석하였다. 첫째, Cougar와 TinyDB는 다중 질의처리를 지원하지 못하며, Agilla는 다중 에이전트들을 이용하여 부분적으로 다중 질의처리를 지원한다. 이에 비해 SwissQM과 BDSM은 최대 6개의 질의를 동시에 수행 가능하다. 둘째, Cougar와 TinyDB는 SQL과 유사한 질의어를 지원함으로써, 질의 전송 메시지 수가 증가하여 통신비용이 증가하는 문제점이 존재한다. 반면, Agilla, SwissQM, BDSM은 바이트코드 형태의 질의어를 지원함으로써, 질의 수행에 필요한 메시지의 전송비용을 감소시킨다. 셋째, 제한된 에너지를 효율적으로 사용하기 위한 추가적인 데이터 집계 기법으로써 BDSM은 Top-K 질의처리와 데이터 필터링 기법을 지원한다. 마지막으로, 기존 미들웨어들은 특정 센서 노드 플랫폼에 의존적으로 개발되어 센서 노드 플랫폼에 따라 미들웨어 사용 가능성이 결정되는 문제점이 존재한다. Cougar와 Agilla는 각각 Mica와 Mica2에 의존적으로 개발되었으며, TinyDB와 SwissQM은 Mica2의 특정 센서 모듈과 TelosB에서만 동작한다. 이에 비해 BDSM은 각 플랫폼에

적합한 모듈만을 연결함으로써 다양한 하드웨어 플랫폼에 대한 추상화를 지원한다.

표 7. 미들웨어 비교 분석

	Cougar	Agilla	Tiny DB	SwissQM	BDSM
다중 질의 처리	×	△	×	○ (6개)	○ (6개)
질의문 형태	SQL	Byte code	SQL	Byte code	Byte code
추가 데이터 집계 질의 처리	×	×	×	×	○ (Top-K, 데이터 필터링)
하드웨어 추상화	×	×	△	△	○

○ : 지원, △ : 부분 지원, × : 미지원

5. 결론 및 향후 연구

본 논문에서는 지오센서 네트워크에서 에너지 효율성을 지원하는 바이트코드 기반 데이터 서비스 미들웨어를 제안하였다. 제안하는 데이터 서비스 미들웨어는 첫째, 하드웨어 추상화를 지원하는 데이터 관리자와 다양한 하드웨어 플랫폼에서 동작할 수 있는 최소한의 바이트코드 명령어를 통하여, 센서 노드에 최적화된 기능만을 제공한다. 둘째, 불필요한 데이터를 제거함으로써 데이터 전송량을 최소화하는 데이터 집계 질의처리 및 데이터 필터링을 통하여, 센서 노드의 에너지 효율성을 향상시킨다. 아울러, 가상 머신 기반 미들웨어인 SwissQM과의 성능 비교를 통하여 데이터 집계 질의처리 수행 시 데이터 전송량이 최대 16%, 평균 에너지 소모량이 최대 37% 감소되며, 대표값 필터링을 적용할 경우 데이터 전송량 측면에서 최대 6배 성능이 향상됨을 알 수 있었다.

향후 연구로는 다양한 응용 서비스를 효율적으로 지원할 수 있는 응용 서비스 관리자의 추가적인 개발 및 본 논문에서 개발한 바이트코드 기반 데이터 서비스 미들웨어가 실제 대규모 지오센서 네트워크에서 효율적임을 보이는 것이다.

참 고 문 헌

- [1] 김호, 백성하, 이동욱, 김경배, 배혜영, 2010, “GeoSensor 환경에서 공간 질의 정확도 향상을 위한 선-필터링을 이용한 후-부하제한 기법”, 한국공간정보시스템학회 논문지, 제12권 제1호, pp. 18-27.
- [2] 유기현, 남광우, 2008, “공간 데이터스트림을 위한 조인 전략 및 비용 모델”, 한국공간정보시스템학회 논문지, 제10권 제4호, pp. 59-66.
- [3] S. Madden et al., 2005, “TinyDB: an acquisitional query processing system for sensor networks”. ACM Trans. Database Systems, vol.30, no.1, pp. 122-173.
- [4] R. Müller, G. Alonso, and D. Kossman, 2007, “A Virtual Machine For Sensor Networks”, ACM SIGOPS/EuroSys European Conference on Computer Systems, vol.41, no.3, pp. 145-158.
- [5] T. Lindholm and F. Yellin, 1998, The Java Virtual Machine Specification, 2nd Ed., Addison-Wesley Professional.
- [6] Y. Yao and J. Gehrke, 2002, “The Cougar Approach to In-Network Query Processing in Sensor Networks”, SIGMOD Record, vol. 31, no. 3, pp. 9-18.
- [7] C. Fok, G. Roman, and C. Lu, 2005, “Agilla: A Mobile Agent Middleware for Sensor Networks”, IEEE Information Processing in Sensor Networks, pp. 382-387.
- [8] S. Madden, M. J. Franklin, J.M. Hellerstein, and W. Hong, 2003, “Tag: A Tiny Aggregation Service for ad hoc Sensor Networks”, ACM Transactions on Database Systems, vol.30, no.3, pp. 122-173.
- [9] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, 2000, “System architecture directions for networked sensors”, ACM SIGPLAN, vol.35, no.11, pages 93 - 104.
- [10] W. Minji, X. Jianliang, T. Xueyan, and L.Wang-Chien, 2007, “Top-k Monitoring in Wireless Sensor Networks”, IEEE Trans. Knowledge and Data Engineering, vol.19, no. 7, pp. 962-976.
- [11] K. Elleithy, T. Sobh, A. Mahmood, M. Iskander and M. Karim, 2006, “Decentralized Kalman Filter in Wireless Sensor Networks - Case Studies”, Advances in Computer, Information, and Systems Sciences, and Engineering, pp. 61-68.
- [12] A. Woo, T. Tong and David Culler, 2003, “Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks”, ACM Conference on Embedded Networked Sensor Systems, pp. 14 - 7.
- [13] P. Levis, N. LEE, M. Welsh, and D.Culler, 2003, “TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications”, ACM Conference on Embedded Networked Sensor Systems, pp. 126-137.
- [14] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, 2003, “The nesc language: A holistic approach to networked embedded systems”, Programming Language Design and Implementation, pp. 1-11.

논문접수 : 2010.02.19

수정일 : 2010.08.06

심사완료 : 2010.09.29



홍 승 태

2008년 전북대학교 컴퓨터공학과
(공학사)

2010년 전북대학교 컴퓨터공학과
(공학석사)

2010년~현재 전북대학교 컴퓨터공학

과 박사과정

관심분야는 공간 데이터베이스, 센서 네트워크, 클라우드 컴퓨팅



윤 민

2008년 전북대학교 컴퓨터공학과
(공학사)

2010년 전북대학교 컴퓨터공학과
(공학 석사)

관심분야는 공간 데이터베이스, 센서

네트워크



장 재 우

1984년 서울대학교 전자계산기공학과
(공학사)

1986년 한국과학기술원 전산학과
(공학석사)

1991년 한국과학기술원 전산학과

(공학박사)

1996년~1997년 Univ. of Minnesota, Visiting Scholar

2003년~2004년 Penn State Univ., Visiting Scholar.

1991년~현재 전북대학교 컴퓨터공학과 교수

관심분야는 공간 네트워크 데이터베이스, 허부지장구조,
센서 네트워크