

다중 이기종 센서를 보유한 Nano-Q+ 기반 센서네트워크에서 메타데이터 라우팅 테이블을 이용한 질의 최적화

(Query Optimization with Metadata Routing Tables on Nano-Q+ Sensor Network with Multiple Heterogeneous Sensors)

남 영 광 [†] 최 귀 자 ^{**} 이 병 대 ^{***}
(Young-Kwang Nam) (Gui-ja Choe) (Byoung-Dai Lee)

곽 광 웅 ^{**} 이 광 용 ^{****} 마 평 수 ^{****}
(Kwang-Woong Kwak) (Kwang-Yong Lee) (Pyoung-Soo Mah)

요 약 일반적으로 센서노드간의 데이터통신은 내부처리나 센싱 작업보다 더 많은 에너지 소모를 요구한다. 본 논문에서는, 내용인지(context-aware) 라우팅 테이블(routing table)을 이용하여 인접한 노드간의 질의 송수신을 위해 필요한 패킷 송신 수를 줄여 질의 최적화를 수행하는 새로운 아이디어를 제안한다. 내용인지 라우팅 테이블에는 현재 노드로부터 도달 가능한 하위노드에서 측정할 수 있는 센서의 종류에 관한 정보가 저장되어 있다.

내용인지 라우팅 정보를 이용하여 각 노드는 자식노드에게 불필요한 질의 송신이나 결과 전달을 차단함으로써 불필요한 패킷 송신의 수를 줄일 수 있다. 본 논문에서 제안한 방법을 바탕으로 한 시뮬레이션에서 최대 약 80%의 성능 효과를 보였다.

키워드 : 센서네트워크, 질의최적화, 메타데이터, 라우팅 테이블

Abstract In general, data communication among sensor nodes requires more energy than internal processing or sensing activities. In this paper, we propose a noble technique to reduce the number of packet transmissions necessary for sending/receiving queries/results among neighboring nodes with the help of context-aware routing tables. The important information maintained in the context-aware routing table is which physical properties can be measured by descendent nodes reachable from the current node. Based on the information, the node is able to eliminate unnecessary packet transmission by filtering out the child nodes for query dissemination or result relaying. The simulation results show that up to 80% of performance gains can be achieved with our technique.

Key words : Sensor Networks, Query Optimization, Metadata, Routing Table

[†] 종신회원 : 연세대학교 컴퓨터정보통신공학부 교수
yknam@yonsei.ac.kr

^{**} 비 회 원 : 연세대학교 컴퓨터정보통신공학부
gchoe@hosu.yonsei.ac.kr
mizzuo@yonsei.ac.kr

^{***} 비 회 원 : 삼성전자 차세대단말기연구단
byoungdai.lee@samsung.com

^{****} 비 회 원 : 한국전자통신연구원 임베디드소프트웨어 연구단 선임연구원
kylee@etri.re.kr
psma@etri.re.kr

논문접수 : 2007년 9월 6일
심사완료 : 2008년 1월 8일

Copyright©2008 한국정보과학회: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 테더 제14권 제1호(2008.2)

1. 서론

센서네트워크는 한 가지 종류의 센서만을 사용하는 것이 아니라 각 센서노드마다 여러 종류의 센싱 장치를 부착하여 네트워크를 구성하는 것이 가능하다. 다기능 센서노드는 온도, 습도, 그리고 초음파와 같은 서로 다른 물리적 특성을 동시에 센싱할 수 있다. 이와 동시에 센서노드는 센싱된 데이터를 이웃한 노드에게 보내기 전에 내부적으로 처리할 수 있는 능력을 가지고 있다. 그러한 환경 하에서 센서네트워크 사용자들이 센서사용 환경에 대한 물리적 특성을 효율적, 효과적으로 조사하기 위해서는 다양한 종류의 질의가 지원되어야 한다.

센서네트워크에서 질의는 서버에서 보낸 질의가 단말 센서노드에 전달되고, 단말 센서노드로부터의 센서데이터가 서버까지 전달되어 질의엔진에서 처리되기 위해서는 센서노드에서 여러 단계의 명령어 처리 과정을 거친다. 이 과정은 네트워크 형상에 관계없이 각 센서노드에서는 질의 송출(query dissemination)과 질의 처리(query processing)의 두 단계로 나눌 수 있다. 질의 송출은 부모노드로부터 받은 질의 명령 패킷을 자식노드에게 전달하는 과정과 질의 결과를 부모노드에게 전달하는 과정으로 나뉘어진다. 이 때 센서노드에 저장된 라우팅테이블은 데이터를 송신하기 위한 적절한 부모노드와 자식노드를 찾는데 중요한 역할을 한다. 질의 처리는 요청 받은 물리적 특성을 센싱하는 활동과, 필요할 경우, 인접한 센서노드와 통신하기 전에, 필터링(filtering) 또는 집합 함수(aggregate function)에 대한 처리 등과 같은 내부 처리 작업을 포함한다.

센서 네트워크에서의 질의 최적화는 관계형 데이터베이스에서 조인 연산의 수를 줄이거나 혹은 연산 순서를 변경하여 테이블 접근 횟수를 줄이도록 하는 질의 최적화와는 달리 에너지의 손실을 최소화하는 것에 중점을 두고 있다[1]. 센서노드 자체는 메모리의 크기가 작고 CPU의 성능이 떨어지기 때문에 복잡한 최적화 프로그램을 적재하는 것이 불가능하며 적재한다하더라도 센서노드의 수명이 다하면 센서노드의 회수가 거의 불가능하기 때문에 가능한 한 노드의 수명을 연장하는 방법으로 최적화가 이루어진다.

센서노드는 노드 자체 내에서의 연산 혹은 센싱에 의한 전력소모 보다는 노드 간에 정보를 전달할 때 가장 많이 전력을 소모하는 것으로 알려져 있다[2]. 따라서 센서노드 질의 처리에서 사용되는 최적화 방법은 전력을 가장 많이 소모하는 질의 송출 단계에서 질의 명령어와 질의 결과를 보내는 횟수를 줄이는 것이 최선의 방법으로 알려져 있다. 센서노드의 내용인지(context-aware) 라우팅 테이블(routing table)은 부모와 자식노

드에 관한 에너지 효율을 고려한 패킷 라우팅정보를 추적하고, 현재노드로부터 도달 가능한 하위노드에 관한 능력정보를 포함한다(이하, 하위노드의 능력에 관한 정보를 메타데이터(metadata)라고 정의함). 각 센서노드는 라우팅 테이블을 이용하여 자기의 자식노드에게 질의를 송출할 필요가 있는지를 결정할 능력을 가진다. 예를 들어, 현재 노드의 자식노드 A로부터 도달 가능한 모든 하위노드가 온도를 측정할 능력이 없을 때, 노드 A에게 온도를 포함하는 질의를 송출할 필요가 없다. 이와 같은 내용인지 라우팅 테이블을 MRT(Metadata Routing Table)이라고 한다. MRT를 유지해야 하는 센서노드의 위치 및 종류에 따라서 MRT를 FMRT(Full Metadata Routing Table)와 SMRT(Sparse Metadata Routing Table)의 두 가지 방법으로 생성한다. FMRT 방법은 센서네트워크 구성에 참여한 모든 노드가 MRT 정보를 가지도록 하는 방법이고, SMRT 방법은 센서네트워크에 동적으로 추가/삭제되는 노드에 관한 MRT 정보를 유지하는데 드는 비용을 고려하여 센서네트워크의 일부 노드만이 MRT정보를 유지하도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 질의 최적화와 관련된 연구를 비교 검토하고, 3장에서는 Nano-Q+ 시스템에서 구현된 질의 처리 시스템의 구조에 대해서 설명한다. 4장에서는 FMRT와 SMRT를 생성하는 방법에 대해서 설명하고, 5장에서는 제안된 방법에 대하여 구현된 결과와 시뮬레이션에 의한 성능 향상에 대하여 기술한다.

2. 관련연구

본 연구와 가장 비슷한 질의 최적화 연구로서는 SRT(Semantic Routing Tree)를 이용한 방법이 있다 [3]. SRT 방법은 각자 노드가 가지고 있는 값의 특성을 부모노드에 저장하여 질의가 주어질 때 자식노드에서 제공되는 값이 질의에서 원하는 의미의 값에서 벗어날 경우에는 자식노드에게 질의를 송출하지 않도록 하는 방법이다. 따라서 SRT에는 항상 자식노드에 대한 특성을 저장하고 있어야 한다. SRT의 성능은 20×20 그리드 내에서 노드의 위치정보를 메타데이터로 이용하여 실제로 측정해본 결과 최대 약 80%의 성능 향상을 보였으며 질의의 값의 범위가 커짐에 따라 선행적으로 더 많은 노드를 포함한다. 이 방법을 좀 더 발전시킨 방법로서는 DSRT(Dynamic Semantic Routing Tree)가 있다[3]. 이 방법은 일정 기간 동안의 값을 저장하고 계속해서 그 값을 갱신하여 항상 최신의 값을 비교하여 그 값의 범위에 들면 질의를 처리하도록 하는 방법이다. [4]에서 제시한 방법은 다중 집합 함수의 계산이 필요한 질의에서 계산 결과를 공유하는 방법을 통해서 통신비

용을 최소화하도록 하였다. 이러한 방법들은 모두 센서 네트워크의 모든 노드들이 같은 종류의 센서로 구성된 상태에서 최적화를 다룬 방법이다.

[5]에서는, 각 노드는 지역 노드의 센서 정보를 교환하여 전역 데이터에 의해 추정되는 이벤트를 감지할 수 있다. 네트워크 범위 내에서 데이터를 감지하는 경우, 중앙 집중 방식이 구현하기는 쉽지만 통신에 대한 부담이 따른다. 이러한 문제는 지역 센서 노드 사이에서 데이터를 교환함으로써 통신비용을 줄일 수 있다. TinyDB에서는 주소-중심 방식에 비해 데이터-중심 방식이 더 효과적이라고 제안한다[2]. [6]에서는 베이스 스테이션을 루트로 하여 라우팅 트리를 반대 경로로 전송할 수 있는 간단한 스키마를 제안하고 있다. [7]은 데이터-중심 저장 스키마를 사용하여 질의를 전송하는 방법을 제안하고 있다. 데이터-중심 방식은 공간적인 중복 혹은 이기종 센서의 임의 배치를 위한 네이밍 스키마를 어렵게 한다. 데이터-중심 라우팅 방식에서의 네이밍 스키마는 주소-중심 라우팅 방식보다 연관성이 적다. [8]에서는 센서 데이터의 내용에 따라 특정 데이터 저장 지점을 생성하여 센서 네트워크를 지리적인 해쉬 테이블 형태로 모델화 하였다. 이 방법은 데이터 검색 작업을 크게 줄여 준다. 추상 영역은 데이터 공유를 위한 원시적 통신 세트를 지원하기 위한 지역성과 관련이 있다[9]. 이것은 프로그래밍, 제어 및 통신을 함께 묶는 원시 세트에 초점을 맞춘다. [10]에서는, 노드의 위치와 지역 토폴로지에 관한 정보를 이용하여 최적 경로를 선택해서 데이터를 전송하는 분산 방식을 제안하였다. 즉, 데이터-중심 접근 방식을 채택하여 분산 알고리즘을 지원하는 방식으로, 다중 라우팅 기법을 제공하는 것을 목표로 한다.

3. Nano-Q+ 기반 질의 처리 시스템

3.1 Nano-Q+ 시스템

Nano-Q+ 시스템은 확장 및 재구성 가능한 시스템으로서 Nano-OS 운영체제와 센서 하드웨어로 구성되어 있다. Nano-Q+ 스택은 태스크 스케줄러, 나노-HAL, 전력관리기, RF 메시지 처리기와 같은 여러 핵심 모듈로 구성되어 있다. 그림 1에 Nano-OS의 핵심 모듈 구조가 나타나 있다.

센서 하드웨어는 그림 2와 같이 저비용, 저전력, 고도의 모듈화에 초점을 맞추어서 만들어진 것으로서 메인블록(Main block), 기본블록, 센서모듈, 활성화기(actuator) 모듈로 구성되어 있다. 메인블록은 ATmega128 마이크로 컨트롤러와 CC2420 IEEE 802.15.4 RF Transceiver를 가진다. 기본 블록은 RS-232 시리얼 인터페이스와 병렬 I/O, 그리고 전원장치로 구성되어 있다. 센서모듈은 온도, 습도, 온도와 Ultra를 측정하는 센서를 가질 수 있다. 활성화기 모듈은 전기스위치와 기타 장치로 구성

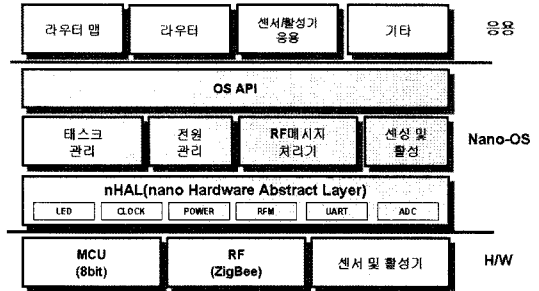


그림 1 Nano-OS 시스템 구조도

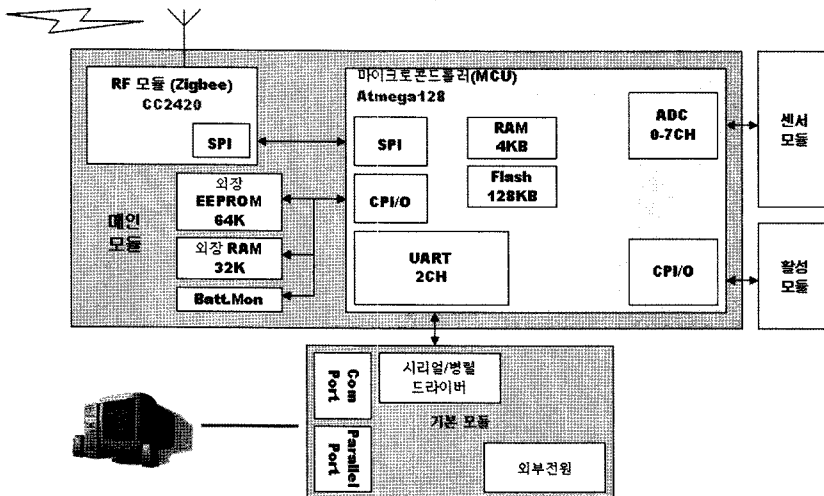


그림 2 Nano-Q+ 시스템 하드웨어 구조도

된다. 센서노드는 두 개의 AA 건전지를 사용한다.

Nano-Q+ 플랫폼의 확장성을 위해서 네트워크 스택은 Nano-OS 위에 리눅스 기반 C 코드로 구현하였다. 네트워크 구조는 계층 모델로 구성되어 있으며 라우팅 테이블과 애드혹(ad-hoc) 노드의 수를 최소화하기 위해서 클러스터-헤더 기반의 스타-매쉬 네트워크 구조를 사용하였다. 각 모듈의 기능은 다음과 같다.

MAC : MAC 계층의 주요 기능은 충돌 방지를 위한 노드 접근을 제어하는 것이다. CSMA-CA에 기반을 둔 IEEE 802.15.3의 일부분을 구현하였다. 센서통신을 위해서 에너지 절약과 조정 기능에 중점을 두었다.

라우팅 : 센서네트워크에서의 라우팅은 인터넷 혹은 애드혹 무선 네트워크에서의 라우팅과 다르다. 예를 들어 센싱 데이터는 싱크노드에 의해서만 전달된다. 또한 작동 명령은 싱크노드로부터 활성기 노드로 전달되도록 설계하여 네트워크를 운영하기 위한 메모리와 전력의 사용을 절감할 수 있도록 하였다.

소켓 API : 센서노드의 메모리가 매우 적기 때문에 TCP나 UDP와 같은 소켓 API를 운영체제의 일부로서 제공한다.

패킷 형태 : 계층적 모델에서, 다른 계층에 대해 독립적으로 구현되어 있다. 각 계층은 헤더 필드의 내용을 채운 후에 하위계층으로 보낸다.

Rx/Tx 버퍼 : 제한된 네트워크 스택에는 입출력되는 패킷을 처리하기 위해서 다섯 개의 배열 포인터를 가진 독립적인 Rx/Tx 버퍼를 가지고 있다.

3.2 NanoDB (Nano-Q+ 질의처리 시스템)의 구조

Nano-Q+ 기반에서의 질의처리 시스템은 NanoDB라 한다. NanoDB 시스템의 구조는 그림 3과 같다. PC 서버에서는 사용자를 위한 GUI, 질의 파서, 명령어 생성기(command generator), 명령어 송출기(command dis-

seminator), 결과 변환기, 질의 처리기 등으로 구성된다. 사용자 GUI에서는 그래픽 기반 GUI와 사용자가 직접 질의를 입력할 수 있는 텍스트 기반 GUI가 제공된다. 사용자는 원하는 질의 조건만 입력하면 결과가 GUI 화면에 출력된다. 텍스트 기반 GUI에서는 질의를 XQuery로 입력하면 결과가 출력 창에 출력된다. 질의 파서는 질의를 파싱하며, 명령어 생성기는 파싱된 질의를 기반으로 센서노드에게 전달한 명령어 패킷(command packet)을 생성한다. 생성된 명령어 패킷은 센서노드가 인식할 수 있는 패킷으로 변환되고 각 질의에 적합한 패킷으로 변환되어 명령어 송출기에 의해 각 센서노드에 전달된다. 결과 변환기는 센서노드에서 측정된 데이터 패킷을 질의 처리에 적합한 XML 형태로 변환하는 역할을 한다.

서버로부터 명령어 패킷을 전달받은 센서노드는 각 노드가 가지고 있는 라우팅 테이블을 이용하여 부모로부터 받은 명령어를 자신의 자식노드에 맞게 재생성하여 명령어를 보낸다. 명령어가 단말노드까지 전달되면 단말 노드는 센싱한 데이터를 부모노드로 보내는데 이때 라우팅 테이블에 있는 부모노드의 정보를 이용하여 센싱 결과를 보낸다. 내부 센서노드는 하위 자식노드로부터 전달받은 질의 결과를 해석하여 외장 플래시 메모리에 저장하고 모든 자식노드로부터 결과를 수신한 경우 이를 부모노드로 전달한 뒤 전력 소비를 최소화하기 위해 휴식 모드로 전환된다.

4. MRT 기반 질의 최적화

4.1 FMRT

NanoDB 시스템에 있는 하나의 센서노드가 명령어 패킷을 받을 때, 그 노드의 모든 자식노드에게 명령어 패킷을 보내지 않는다. 대신, 자식 노드의 센싱 능력과 자식 노드에서 도달할 수 있는 하위노드의 부분집합에

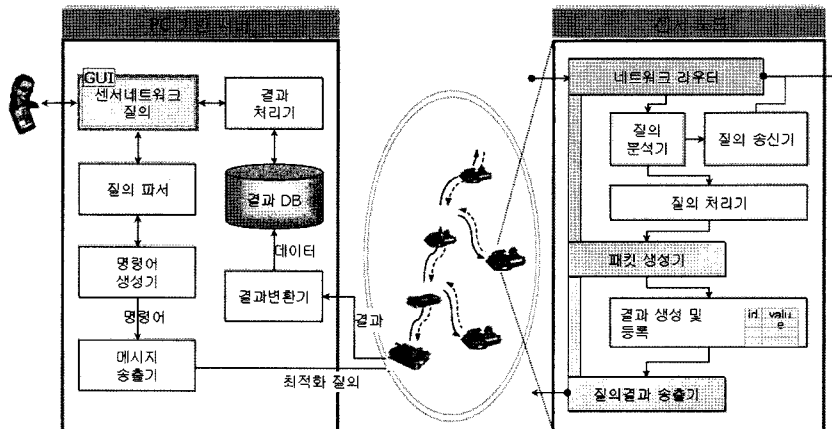


그림 3 NanoDB 시스템의 구조

게만 패킷을 보낸다. 하위노드의 능력에 관한 정보를 메타데이터라 하고 MRT에 그 정보를 저장한다.

설명을 간결하게 하기 위하여 다음과 같이 정의한다. 센서네트워크 $SN = \{S_1, S_2, \dots, S_n\}$ 에는 n 개의 센서노드가, $SR = \{R_1, R_2, \dots, R_m\}$ 은 각 센서노드 S_i ($1 \leq k \leq n$)가 가질 수 있는 메타데이터를 의미한다. 추가로 참여하는 각 노드마다 이용 가능한 센서타입의 집합 S_i 는 서로 다르지만 SR 의 부분집합이다.

i) $C_i = \{C_i^1, C_i^2, \dots, C_i^p\}$ 를 S_i 의 자식노드의 집합이라 하고 C_i^k 는 S_i 의 k 번째 노드를 의미한다.

ii) $A_i = \{B_i(R_1), B_i(R_2), \dots, B_i(R_m)\}$ 은 S_i 의 능력테이블이라 하며, $B_i(R_j)$ 는 만약 S_i 가 R_j 를 센싱할 수 있으면, 참값을 가진다.

iii) $M_i^k = \{M_i^k(R_1), M_i^k(R_2), \dots, M_i^k(R_m)\}$ 는 C_i^k 와 관계된 메타데이터라 하고, $M_i^k(R_j)$ 값은 a) C_i^k 를 통하여 S_i 에 도달할 수 있거나, b) R_j 를 센싱할 수 있으면 참값을 가지고 그렇지 않으면 거짓값을 가진다.

iv) $M_i = \{M_i^1, M_i^2, \dots, M_i^p\}$ 은 S_i 에 저장되어 있는 메타데이터의 집합이며, p 는 S_i 의 자식노드의 수를 의미한다.

센서타입의 집합 $QR = \{R_1', R_2', \dots, R_p'\}$ 을 포함하는 질의가 주어졌을 때, 질의 송출을 위한 노드선택은 A_i 값에 의해서 결정됨을 쉽게 알 수 있다. 다음은 노드 S_i 에서 실행되는 알고리즘이다.

```

① Let  $T = \emptyset$ 
② for each child node  $C_i^k$  in  $C_i$ , where  $k = 1..p$ 
   for each  $M_i^k(R_j)$  in  $M_i^k$ , where  $j = 1..m$ 
③   if ( $M_i^k(R_j) == true$ ) and ( $R_j \in QR$ )
④      $T = T \cup C_i^k$ 
⑤

```

만약 자식노드 C_i^k 를 가지는 서브트리가 루트노드로서 질의에 관련이 있는 물리적 특성을 센싱할 수 있다면, C_i^k 는 질의 송출에 대한 목적지로서 포함된다.

FMRT 방법에서는, 센서네트워크에 있는 모든 참여노드는 MRT를 가진다. 각 노드는 자체의 MRT를 만들기 위해서 자식노드로부터 메타데이터의 정보를 받는다. 그 정보를 이용하여 현재노드와 그것의 하위노드에 관한 능력정보가 부모노드로 전달된다. 이러한 과정은 하나의 센서노드에 저장된 메타데이터가 그것의 모든 하위노드에 관한 상태를 반영하기 때문에 루트노드의 MRT가 생성될 때까지 계속된다. 다음은 MRT를 수정하기 위한 알고리즘이다.

```

①  $V_i = \{V_i(R_1) = false, V_i(R_2) = false, \dots, V_i(R_m) = false\}$ 
② for each  $R_k$  in  $SR$ , where  $k = 1..m$ 
③    $V_i(R_k) = B_{i,t}(R_k) \parallel (M_{i,t}^1(R_k) \parallel M_{i,t}^2(R_k) \parallel \dots \parallel M_{i,t}^p(R_k))$ 
④  $V_{i+1} = \{V_{i+1}(R_1) = false, V_{i+1}(R_2) = false, \dots, V_{i+1}(R_m) = false\}$ 
⑤ for each  $R_k$  in  $SR$ , where  $k = 1..m$ 
⑥    $V_{i+1}(R_k) = B_{i,t+1}(R_k) \parallel (M_{i,t+1}^1(R_k) \parallel M_{i,t+1}^2(R_k) \parallel \dots \parallel M_{i,t+1}^p(R_k))$ 
⑦ if  $V_i \neq V_{i+1}$ , send  $V_{i+1}$  to the parent nodes

```

①-③까지는 t 시점에 부모노드에 보내진 메타데이터를 계산한다. 그러나 이 정보는 새로운 메타데이터가 부모노드에 보내질 때까지만 유지될 수 있다. 부모노드에 전달된 메타데이터는 현재노드가 루트노드 역할을 하는 서브트리에 대한 능력을 나타내기 때문에, 현재 노드의 능력도 반드시 ③에서 고려되어야 한다. 줄 ④-⑥은 현재노드의 새로운 능력테이블과 하위노드의 능력에 기반을 둔 새로운 메타데이터를 계산한다. 새로운 메타데이터는 두 메타데이터집합이 다를 때, 즉 부모노드의 관점에서 루트노드로 S_i 를 가지는 서브트리가 축적할 수 있는 물리적 특성의 수가 증가되거나 감소된다는 것을 의미한다. 다시 말하면, 만약 서브트리에 있는 노드들의 형태의 변경이 전체 서브트리에 대한 능력의 변경을 일으키지 않으면, 부모노드의 MRT 값을 최신의 정보로 유지하기 위한 추가의 비용이 요구되지 않는다.

4.2 SMRT

MRT의 중요한 기능은 패킷 송출수를 줄이기 위해 현재 노드에서 질의에 포함되지 않은 센서를 가지고 있는 서브트리를 식별해줄 수 있도록 하는 것이다. FMRT 방법은 질의 송출과 결과 전달에 필요한 패킷의 수를 줄여줄 수 있지만, 이 방법은 MRT를 유지하기 위해서 추가의 계산과 통신이 필요하고, 메모리를 많이 사용하며, 네트워크의 모든 노드를 방문해야하기 때문에 비효율적이고 비용이 많이 들 수 있다. 특히, 시간의 흐름에 따라 네트워크를 들고 나는 센서노드가 많을 경우에 현재의 MRT를 유지하기 위한 비용이 많이 증가한다. 이러한 문제점을 해결하기 위해서 SMRT 방법에서는, 센서네트워크의 노드 구성이 변할 경우 불필요한 메타데이터의 전달을 제거하여 성능을 향상시키도록 FMRT 방법을 수정하였는데, R_j 를 센싱할 수 있는 능력을 가진 노드만이 R_j 에 대한 하위노드에 대한 메타데이터 정보를 유지할 수 있도록 하였다. 만약 이 조건이 만족되지 못하면, 그 노드는 자신의 MRT를 제거하고 그 사실을 모든 자식노드에게 알린다. 이와 같은 방법으로 MRT를 동적으로 유지하여, MRT를 가진 노드의 수를 최소화한다. 다음은 MRT 갱신을 위해서 부모노드 S_i 에서 실행되는 알고리즘을 나타낸다. 설명을 간결하게 하기 위해서 각 노드는 같은 종류의 단일 센서를 가졌다고 가정한다.

```

① In case that a new node,  $C_j$  is added into the network
②   if  $V_{j,t}(R) = true$ 
③     remove an entry  $M_{i,t}^j(R)$  in  $M_{i,t}^j$ 
④     send  $V_{i,t}(R) = true$  to the parent nodes
⑤   else
⑥     set  $M_{i,t}^j(R) = false$ 
⑦ In case that a node,  $C_j$  is removed from the network

```

```

(8) if  $V_{j,t}(R) = false$ 
(9)   do nothing
(10) if  $V' = (M_{j,t+1}^1(R) \| M_{j,t+1}^2(R) \| \dots \| M_{j,t+1}^y(R)) = false$  and
       $B_{j,t}(R) = true$ , where  $y$  is the number of child nodes of  $C_j$ 
(11)   set  $M_{j,t}^y(R) = false$ 
(12)   send  $V_{j,t}(R) = false$  to the parent nodes
(13) else if  $V_{j,t}(R) = true$  and  $B_{j,t}(R) = false$ 
(14)   do nothing
    
```

만약 새로운 노드와 그것의 하위노드가 R_j 를 측정할 수 있다면, 새로운 노드가 R_j 를 처리하기 위해서 포함되어야 하기 때문에 조상노드들은 R_j 에 대한 MRT를 제거한다. 이러한 변경은 모든 조상노드들에게 전달된다. 반대로 새로운 노드와 새로운 노드의 서브트리가 R_j 를 측정할 수 있는 능력을 가지지 않았다면 부모노드에 대한 MRT는 질의가 보내지지 않도록 수정되어야 한다(⑤-⑥). 이러한 갱신은 조상노드의 MRT에 영향을 미치지 않는다. 어떤 노드가 센서네트워크에서 삭제될 경우 삭제될 노드가 R_j 를 측정할 수 있는 노드라면, 부모노드의 MRT가 수정되어야 한다(⑩-⑫). 그렇지 않으면 아무런 변경도 일어나지 않는다.

5. 실험 결과

MRT 방법의 성능을 평가하기 위해서 JProowler[15]를 이용한 센서네트워크 시뮬레이터를 개발하였다. JProowler는 밴더빌트 대학에서 개발한 센서 네트워크 시뮬레이터로서, 최하위 통신 채널에서부터 최상위 응용 프로그램까지 센서노드 및 센서노드 상에서 구동되는 S/W의 다양한 부분에 걸쳐 시뮬레이션이 가능한 이벤트 기반 시뮬레이터이다. 센서 네트워크의 다양한 특성 중에서 JProowler는 크게 네 가지 부분에 대해서 시뮬레이션 모델을 제공한다.

- 1) 신호 전파 모델(Radio propagation models)
- 2) 신호 재전송 및 충돌 모델(Signal repetition and collision models)
- 3) MAC Layer 모델
- 4) 응용 프로그램 모델

본 논문에서 제안한 질의 처리 최적화 방법의 성능을 평가하기 위해, 400개의 노드로 구성된 센서네트워크 상에서 4가지 타입의 센서에 대한 값을 검색하는 질의를 임의로 수행하여 패킷 송신 횟수를 비교하였다. 각 노드마다 가질 수 있는 센서의 종류는 임의로 시뮬레이션 프로그램에서 결정하였다. 그림 4와 그림 5는 네트워크의 트리 깊이와 질의에 포함된 센서종류와 노드의 분포에 따라 임의로 생성된 트리구조를 이용한 FMRT와 SMRT 방법에 대한 성능을 나타낸다. 예를 들어 “FMRT-(4) 80%”는 4가지 센서를 가지고 80% 이상의 노드가 질의를 만족될 수 있는 질의에 대한 FMRT의 성능을 나타낸다. 센서의 종류와 만족하는 노드의 수가 감소할수록 FMRT는 SMRT보다 좋은 성능을 보인다. 그러나 만족되는 노드의 수가 증가됨에 따라, SMRT의 성능이 FMRT와 비슷해진다 것을 알 수 있다. 흥미로운 것은 “SMRT-(1) 10%”와 “SMRT-(1) 50%”에서 만족하는 노드의 수는 감소했지만 트리의 깊이가 커짐에 따라서 성능이 감소하는 것을 알 수 있다. 그 이유는 SMRT에서 MRT를 유지하기 위한 노드의 부분집합과 MRT를 가진 노드의 위치가 전체 성능에 영향을 미치기 때문이다. 예를 들어, MRT를 가진 노드가 트리의 윗부분에 있다면, 불필요한 질의를 송출이 안 될 확률이 높아지기 때문이다.

그림 6은 노드 추가 시에 필요한 송출 횟수를, 그림 7은 삭제 시에 필요한 송출 횟수를 나타낸다. 앞 절의 FMRT와 SMRT 방법에서 알 수 있듯이, FMRT 방법

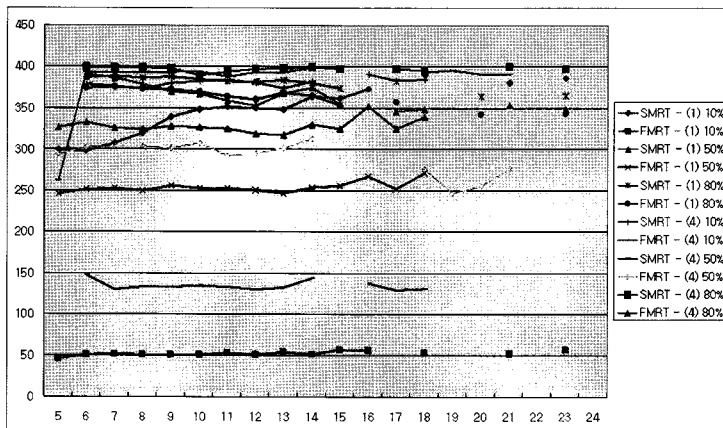


그림 4 Random Tree에 대한 시뮬레이션 결과

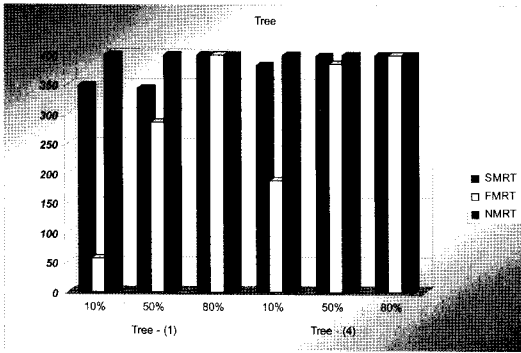


그림 5 센서 포함률, 센서타입별 성능 시험 결과

은 하위노드의 능력여부를 같은 능력이 나타나는 노드까지 전달해야하는 반면에, SMRT는 MRT가 존재하는 노드까지만 전달하기 때문에 추가/삭제시의 비용이 적게 든다.

6. 결론 및 향후 연구 과제

본 논문에서는 Nano-Q+ 시스템을 이용한 센서네트워크에서 질의 검색 시스템을 구현함과 동시에 MRT를 이용한 질의 최적화 방법을 제안하였다. 모든 노드가 MRT를 가질 경우에는 질의에 대한 성능이 향상되지만, 각 노드가 MRT에 대한 정보를 유지해야 하는 부담이

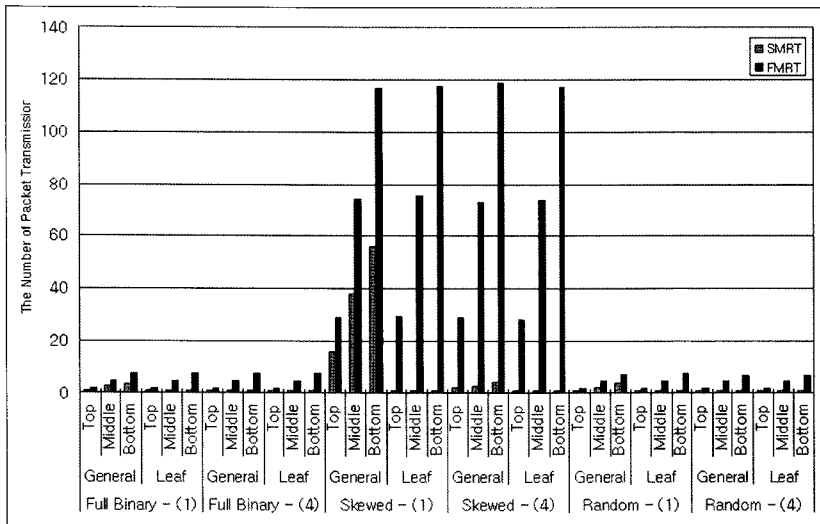


그림 6 노드 추가시 MRT 갱신에 필요한 패킷 전송의 수

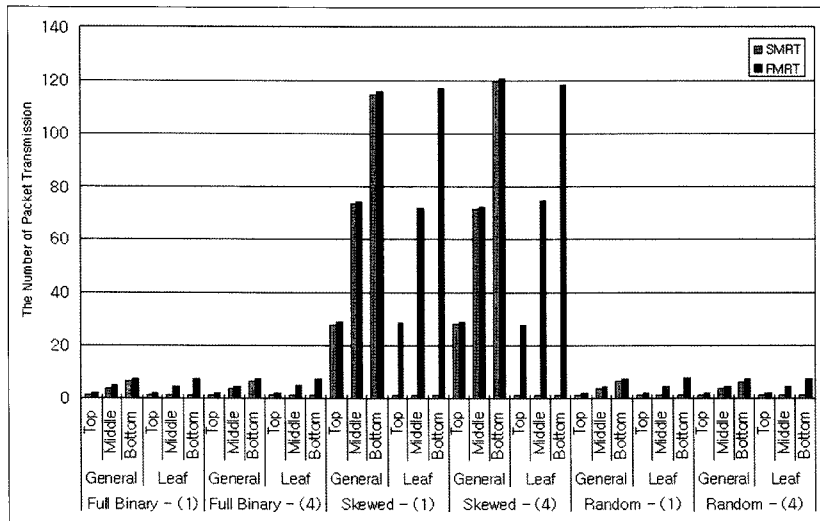


그림 7 노드 삭제시 MRT 갱신에 필요한 패킷 전송의 수

있다. SMRT인 경우에는 FMRT의 단점을 해소하기 위해서 동적으로 최소한의 노드가 노드의 상태가 변경되는 곳에 있는 노드만이 메타데이터 값을 가질 수 있도록 하여 MRT가 없는 노드에 대해서는 무조건 질의를 송출하도록 수정하였다. 이 경우, FMRT보다는 성능이 떨어지지만 모든 노드가 MRT를 유지해야 하는 부담을 줄이도록 하였다. FMRT방법은 센서네트워크에 포함된 센서의 비율에 따라 성능이 선형적으로 증가함을 보였으며, SMRT 경우에는 그 비율이 50%에서 80% 사이 일 경우에 성능이 좋은 것으로 나타났다.

향후 연구 과제로서는 Nano-Q+ 시스템을 위한 시뮬레이터를 이용하여 MRT를 이용한 에너지의 소비량과 MRT를 유지하기 위한 연산 비용을 고려하여 실제로 어느 정도의 에너지가 절약되는지를 알아보고 FMRT와 SMRT를 복합해서 사용하여 어느 정도 깊이에서 FMRT를 사용하고 그 다음부터는 SMRT를 사용하는 것이 좋은 지를 알아내도록 하는 것이다.

참 고 문 헌

- [1] J. Gehrke, S. Madden, Query Processing in Sensor Networks, Pervasive computing, 2004, Jan-Mar.
- [2] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tinydb: An Acquisitional Query Processing System for Sensor Networks. ACM Trans. Database Syst., 30(1):122 - 173, 2005.
- [3] J. Zhu, Dynamic Semantic Routing in Sensor Network, Dept. of CS, RIT, NY, USA.
- [4] J. Shneidman, P. Pietzuch, M. Welsh, et. al, A Cost-Space Approach to Distributed Query Optimization in Stream Based Overlays.
- [5] E.B. Ermis and Venkatesh Saligrama. A Statistical Sampling Methods for Decentralized Estimation and Detection of Localized Phenomena. In Fourth Intl. Symp. on Information Processing in Sensor Networks (IPSN), pages 143-150, April 2005.
- [6] D. Coffin, D. V. Hook, S. McGarry, and S. Kolek. Declarative ad-hoc Sensor Networking. SPIE Integrated Command Environments, 2000.
- [7] Ramesh Govindan. Data-centric Routing and Storage in Sensor Networks. pages 185-205, 2004.
- [8] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. Ght: A Geographic Hash Table for Data-centric Storage. In WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pages 78-87, New York, NY, USA, 2002. ACM Press.
- [9] Matt Welsh and Geoff Mainland. Programming Sensor Networks Using Abstract Regions. In NSDI, pages 29-42, 2004.
- [10] Brad Karp and H. T. Kung. Gpsr: Greedy Perimeter Stateless Routing for Wireless Networks. In MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, pages 243-254, New York, NY, USA, 2000. ACM Press.
- [11] Yong Yao and Johannes Gehrke. The COUGAR Approach to In-network Query Processing in Sensor Networks. SIGMOD Rec., 31(3):9 - 18, 2002.
- [12] A. Agarwal and T.D.C. Little, Attribute Based Routing in Sensor Networks, Boston University, Boston, Massachusetts, MCL Technical Report No. 06-01-2006.
- [13] N. Trigoni, et al., Multi-query Optimization for Sensor Networks.
- [14] B. Krishnamachari, D. Estrin, and S. Wicker. Modeling Data-centric Routing in Wireless Sensor Networks. In IEEE INFOCOM, 2002.
- [15] <http://www.isis.vanderbilt.edu/Projects/nest/jprowler/index.html>



남 영 광

1995년 3월~현재 연세대학교 컴퓨터정보통신공학부 교수. 1993년~1994년 시스템공학연구소 선임연구원. 1992년~Northwestern University 전산학과 졸업. 1985년 한국과학기술원 전산학과 졸업. 1982년 연세대학교 수학과 졸업. 관심분야는 프로그래밍언어, 소프트웨어공학, 정보검색, XML.



최 귀 자

2005년~현재 연세대학교 전산학과 박사과정. 2005년 연세대학교 전산학과(석사) 1991년 서울산업대학교 전산학과(학사) 관심분야는 프로그래밍 언어, XML, 소프트웨어 공학



이 병 대

2003년 10월~현재 삼성전자 정보통신총괄 통신연구소 재직. 2003년 9월 Dept. of Computer Science and Engineering, University of Minnesota, twin cities (박사). 1998년 2월 연세대학교 대학원 전산학과 졸업(석사). 1996년 2월 연세대학교 전산학과 졸업(학사). 관심분야는 소프트웨어공학, 임베디드 S/W



곽 광 응

2006년~현재 연세대학교 전산학과 석사
과정. 2006년 연세대학교 전산학과(학사)



이 광 응

1991년 숭실대학교 전자계산학과(학사)
1993년 숭실대학교대학원 전자계산학과
(석사). 1997년 숭실대학교대학원 전자계
산학과(박사). 1997년~1998년 ETRI 컴
퓨터/소프트웨어기술연구소, 소프트웨어
공학연구부 Post-Doc. 2005년 과학기술
연합대학원대학교(UST) 겸임교수. 1999년~현재 ETRI, 임
베디드S/W연구단 임베디드OS연구팀 선임연구원. 관심분야
멀티코어기술, 임베디드OS(표준/마이크로/나노 운영체제),
유비쿼터스 센서 네트워크(USN), 실시간 디버거/정형기법,
객체지향 실시간 모델링 방법, 소프트웨어공학



마 평 수

1996년~현재 ETRI 임베디드SW연구단,
팀장, 책임연구원. 1995년 Wright State
Univ, Ohio, USA 전산학, 박사. 1992년
City Univ of New York, N.Y., USA
전산학, 석사. 1985년 서울대학교, 식물병
리학, 학사. 관심분야는 센서 네트워크,
센서노드용 OS, 임베디드 SW