

컨테이너 환경에서 토폴로지 재구성 시간을 개선한 변형 LEACH 프로토콜

이 양 민^{*} · 이 기 원^{**} · 광 광 훈^{***} · 이 재 기^{****}

요 약

일반적으로 애드-혹 네트워크에 적용되는 라우팅 알고리즘은 수천 개 이상의 많은 노드를 포함하는 환경에는 적합하지 않다. 이 문제를 해결하기 위해서 노드에 대한 계층적 관리와 토폴로지의 안정적 유지를 위해 클러스터링 기반의 프로토콜을 이용한다. 본 논문에서는 통신 노드들을 장착한 금속 컨테이너들이 이동하는 환경에서도 적용 가능한 클러스터링 기반의 변형 LEACH 프로토콜을 제안한다. 제안한 프로토콜은 클러스터 기반의 프로토콜인 LEACH에 노드 이동을 감지하는 모듈을 구현하였고, 노드가 이동하는 환경에서 LEACH가 가지는 단점을 개선하였다. 그리고 멀티 홉의 구성 방법을 조절하여 효율적인 통신이 가능하도록 하였다. 또한 제안한 프로토콜과 기존의 LEACH 프로토콜을 집진적 네트워크 구성 시간, 토폴로지 재구성 시간, 컨테이너 환경에서의 통신 성공률, 그리고 라우팅 오버헤드라는 네 가지 관점에서 비교하였다. 비교 결과, 본 논문에서 제안한 프로토콜이 금속 컨테이너에 포함된 이동 가능한 노드의 통신에서 기존의 LEACH 프로토콜보다 우수함을 확인하였다.

키워드: 금속 컨테이너 환경, 애드-혹, 프로토콜, 이동 감지

Modified LEACH Protocol improving the Time of Topology Reconfiguration in Container Environment

Yang-Min Lee^{*} · Ki-One Yi^{**} · Gwang-Hoon Kwark^{***} · Jae-Kee Lee^{****}

ABSTRACT

In general, routing algorithms that were applied to ad-hoc networks are not suitable for the environment with many nodes over several thousands. To solve this problem, hierarchical management to these nodes and clustering-based protocols for the stable maintenance of topology are used. In this paper, we propose the clustering-based modified LEACH protocol that can applied to an environment which moves around metal containers within communication nodes. In proposed protocol, we implemented a module for detecting the movement of nodes on the clustering-based LEACH protocol and improved the defect of LEACH in an environment with movable nodes. And we showed the possibility of the effective communication by adjusting the configuration method of multi-hop. We also compared the proposed protocol with LEACH in four points of view, which are a gradual network composition time, a reconfiguration time of a topology, a success ratio of communication on an containers environment, and routing overheads. And to conclude, we verified that the proposed protocol is better than original LEACH protocol in the metal containers environment within communication of nodes.

Key Words: IMetal Container Environment, Ad-hoc, Protocol, Movement Detection

1. 서 론

근래 유비쿼터스 네트워크나 무선 네트워크와 관련

된 연구가 다양하게 진행되고 있다. 이러한 연구에는 여러 분야가 존재하지만 특히 센서를 이용한 정보 수집 목적의 자율적인 애드-혹(Ad-hoc) 네트워크 구성을 위한 프로토콜 연구가 활발하게 진행 되고 있다[1][2].

이러한 네트워크의 특성으로는 센서들이 자율적으로 네트워크를 구성하여 데이터를 수집하는 싱크 노드(Sink Node) 방향으로 통신을 전개할 수 있다는 점 들 수 있으며, 여기에 네트워크를 이루는 노드가 센서

※ 본 연구는 산업자원부 지방기술혁신사업(B00009720) 지원으로 수행되었음.
† 정 회 원: 동아대학교 컴퓨터공학과 시간강사
†† 준 회 원: 동아대학교 컴퓨터공학과 학사과정
††† 정 회 원: 동아대학교 미디어디바이스연구센터 책임연구원
†††† 정 회 원: 동아대학교 컴퓨터공학과 교수
논문접수: 2008년 4월 25일
수정일: 2008년 6월 18일
심사완료: 2008년 7월 8일

를 포함하는 관계로 다수(수 천~수 만)의 노드가 참여하는 대규모 네트워크를 구성한다는 점도 특징으로 볼 수 있다[3].

센서를 이용하여 센싱을 중심으로 하는 에드-혹 네트워크를 구성할 수 있도록 하는 프로토콜들은 크게 두 가지 범주로 나눌 수 있다. 하나는 모든 노드들이 동등한 관계를 구축하여 각각이 베이스 스테이션(Base Station : BS)이 되는 평면적인 기법이다. 다른 하나는 구조적인 기법으로서 클러스터 기반 방식이라고도 하며 통신의 집약 거점으로 활동하는 노드를 선출하여 이 노드가 클러스터 헤드가 되며 하위 노드들의 데이터를 수집, 관리하고 이를 다시 상위 등급의 클러스터 헤드로 전달한다. 보통 노드의 숫자가 많아지면 에너지 소모와 네트워크 토폴로지 관리 등의 이유에서 평면적인 기법의 프로토콜보다는 구조적 기법의 프로토콜을 사용한다[4-6].

근래 들어 에드-혹 네트워크는 적용할 수 있는 응용 범위가 확대되면서 다양한 환경에 활용되고 있다. 예를 들면, 항만과 같은 급속으로 된 컨테이너로 구성되는 환경에서 컨테이너 노드의 이동 고려하고, 급속 환경이라는 특성에서 기인하는 음영 지역을 제거할 수 있는 프로토콜도 요구되고 있다[7-9].

본 논문에서는 클러스터 기반 프로토콜인 LEACH를 급속 컨테이너 네트워크 환경 및 지능형 노드들의 움직임이 존재하는 환경에도 적용할 수 있도록 네트워크 계층 부분을 변형하여 그 성능을 기존의 LEACH 프로토콜과 비교하였다. 네트워크 계층 부분의 기능 변경은 모션 센싱에 의한 빠른 시간 내에 토폴로지를 재구성할 수 있도록 하는 것과 패킷의 이동 자취를 남겨둘 수 있도록 하는 것이다. 또한 급속 환경에서 발생할 수 있는 음영 지역을 제거하기 위해 다른 지능형 노드들을 거쳐 통신이 가능하도록 멀티 홉을 구성할 수 있는 방안을 추가하였다[10-13].

본 논문의 구성은 1장의 서론에 이어 2장에서는 관련 연구에 관해 간략하게 언급하였다. 3장에서는 제안 프로토콜인 변형 LEACH 프로토콜에 대해 기술하고, 4장에서는 기존의 LEACH 프로토콜과 제안 프로토콜과의 비교 실험을 수행한 결과를 나타내었으며 5장의 결론으로 되어 있다.

2. 관련 연구

앞서 기술한 것처럼 센서 기술이 적용된 에드-혹 네트워크에서는 일반적인 에드-혹 라우팅 알고리즘을 적용하기에는 에너지 효율이나 라우팅 정보의 계층적 관리라는 측면에서 문제가 있다[6][7]. 그러므로 관련 연구에서는 에드-혹 네트워크를 구성하는데 있어 대규모의 노드가 포함되는 경우 적용 가능한 센서 네트워크 라우팅 프로토콜에 관해서 간략하게 기술하였다.

2.1 센서 네트워크 라우팅 프로토콜

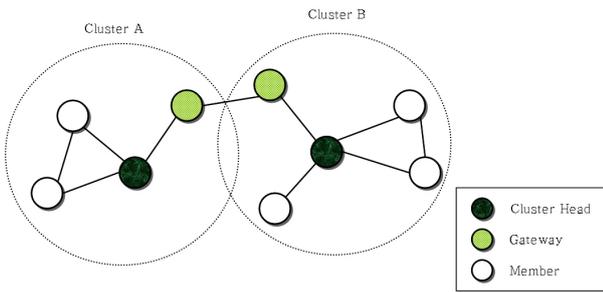
2.1.1 평면 라우팅 프로토콜[1]

다수의 노드가 포함되는 네트워크에서는 속성에 기반을 둔 어드레싱 기법으로 특정 노드의 에너지 소비를 막고 센싱된 데이터의 전달 과정에서 에너지 낭비를 줄이기 위한 데이터 집약을 수행하여야 네트워크의 전체적인 수명을 증가시킬 수 있다. 아래에 대표적인 평면 라우팅 알고리즘을 간략히 기술하였다.

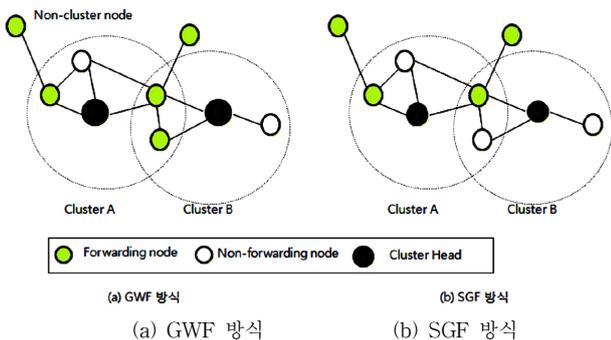
- Directed Diffusion[11] : 싱크의 질의 방송에 기반을 둔 데이터 중심적 라우팅 기법으로 질의 유포 및 처리 응용에 적합한 특성을 가진다. Directed diffusion에서 해당 응용에 대한 질의는 속성과 값의 쌍으로 구성되는 interest로 표현되는데 이는 싱크에서 시작되어 네트워크 전체에 유포되고, 데이터 요청 노드로 질의에 부합하는 데이터를 전송하기 위한 경사가 설정된다. 데이터는 다중 경로를 통해 요청 노드로 전송되는데 이 시점에서 더 이상의 플러딩을 막기 위해 여러 개의 경로들 중 전송 품질이 높은 몇 개의 경로들만 선택된다. 그리고 선택된 경로를 통해 데이터 전송이 이루어진다. Directed diffusion은 지속적인 질의에 적합하나 1회성 질의에는 적합하지 않다.
- Sensor Protocols for Information via Negotiation (SPIN) : SPIN은 협상과 자원 적용에 의해 플러딩의 결함을 처리하기 위해 설계된 것으로, 센서 노드가 데이터에 대해 광고하고, 싱크로부터 요청을 기다리는 형태의 데이터 중심적 라우팅 기법이다. 이는 센서 노드가 직접 데이터를 브로드캐스팅하지 않고 센서 데이터를 기술하는 메타 데이터 전송으로 보다 에너지 효율적이다. SPIN은 ADV, REQ, DATA라는 세 가지 주요 메시지를 가지는데 센서 노드는 메타 데이터를 가지는 ADV 메시지를 브로드캐스팅하고, ADV의 메시지를 수신한 이웃 노드가 데이터에 대한 관심을 가지고 REQ 메시지를 전송하면, 이때에만 해당 이웃 노드에게 실제 DATA 메시지를 전송한다.

2.2 클러스터링 기법[14]

클러스터는 클러스터 헤더라는 하나의 노드와 이 노드로부터 직접 통신 거리, 즉 1 홉 무선 거리에 존재하는 모든 노드의 집합으로서 정의된다. 클러스터 노드 중 게이트웨이는 다른 클러스터의 헤더와 1 홉 거리에 있는 노드, 즉 둘 이상의 클러스터에 속하는 노드로 정의된다. (그림 1)과 같이 클러스터가 서로 중첩되지 않게 구성될 경우, 게이트웨이는 인접 클러스터의 클러스터 구성 노드와 직접 통신하는 노드로 정의된다.



(그림 1) 클러스터링에 기반 센서 네트워크

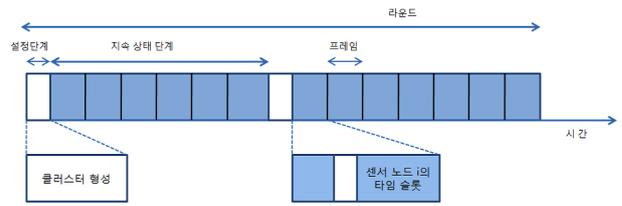


(그림 2) 클러스터링 방식에서의 포워딩 노드 선택

클러스터를 이용한 플러딩을 위해서는 먼저 클러스터를 구성하는 클러스터링이 수행되어야 한다. 클러스터링 방식의 예로 LCA (Linked-Cluster Algorithm)과 HCA(Highest-Connectivity Algorithm)을 들 수 있다[2]. 클러스터링에 의해 클러스터가 구성되었을 경우 GWF(Gateway Forwarding) 방식에서는 (그림 2-a)와 같이 클러스터 헤드, 게이트웨이 및 클러스터에 속하지 않는 노드만이 포워딩 노드가 되어 브로드캐스트 패킷을 플러딩한다. 그러나 게이트웨이 노드들 간의 플러딩은 서로 중복될 가능성이 있기 때문에 클러스터 헤드에 의해 선택된 일부 게이트웨이만 플러딩을 수행할 수 있는 SGF(Selected Gateway Forwarding) 방식을 사용함으로써 플러딩 성능을 향상시킬 수 있다(그림 2-b)[2-4].

아래에 대표적인 클러스터링 기반 라우팅 프로토콜에 대해 간략하게 기술한다.

- Low-Energy Adaptive Clustering Hierarchy (LEACH) : LEACH는 클러스터링 기반 라우팅 기법으로, 클러스터 헤드가 클러스터의 멤버 노드들로부터 데이터를 수집하여 직접 싱크 노드로 전달한다. LEACH의 주요 특징은 네트워크 상에 있는 모든 센서 노드들의 에너지 소비를 공평하게 분산시키기 위해 에너지 소모적 기능을 하는 클러스터 헤드를 무작위로 순환시키고, 전체적인 통신 에너지를 줄이기 위해 클러스터 헤드에서 클러스터내의 데이터를 모아 지역적으로 축약하는 것이다. 이와 같은 목적을 위해서 LEACH의 동작은 (그림 3)와 같은 라운드(Round)라는 시간 단위로 이루어진다.

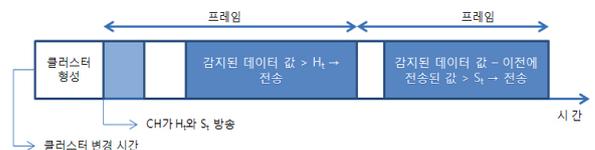


(그림 3) LEACH의 라운드 타임

각 라운드는 클러스터가 구성되는 설정 단계와 여러 개의 TDMA 프레임으로 구성되는 지속 상태 단계로 이루어진다. 설정 단계가 시작되면 모든 노드는 자신이 현 라운드 동안 클러스터 헤드를 수행할 수 있는지에 대해 이전 라운드 동안 클러스터 헤드였는지와 최적의 클러스터 헤드 수에 기반을 두고 결정을 하게 된다. 현 라운드 기간에 클러스터 헤드가 되기로 결정한 노드는 이를 이웃 노드들에 브로드캐스팅하며, 이를 수신한 비클러스터 헤드 노드들은 수신 강도나 에너지 잔류량 등의 인자를 기반으로 자신의 클러스터 헤드를 결정하고 이를 클러스터 헤드로 전송하여 클러스터를 구성하게 된다. 지속 상태 단계에서는 클러스터를 구성하는 멤버 노드들이 자신의 전송 슬롯에서만 데이터를 전송하고 나머지 슬롯들에서는 슬립(Sleep) 모드로 전환하여 전력 소모를 줄인다. LEACH에서는 클러스터 내부에서는 TDMA를 사용하여 노드 간 간섭을 피하고, 각 클러스터 간에서는 확산 코드를 사용하여 클러스터 간의 간섭을 피한다.

LEACH의 성능은 매 라운드마다 일정한 수의 클러스터를 구성하고 클러스터 헤드가 고르게 배치되어야 최적으로 유지되는데, 노드가 스스로를 헤드로 선출하는 방법에서는 이를 보장할 수가 없기 때문에 싱크에서 센서 노드의 위치 정보와 에너지 잔류량을 고려하여 클러스터 헤드와 클러스터를 결정하는 LEACH-C 기법도 존재한다.

- Threshold sensitive Energy Efficient sensor Network protocol (TEEN) : TEEN은 LEACH와 거의 유사하게 동작하지만 센서 노드들이 주기적으로 전송할 데이터를 가지고 있지 않다. LEACH는 사전적 센서 네트워크에 적합하지만, TEEN은 시간을 임계점으로 하여 데이터를 처리하기 때문에 반응적 센서 네트워크에 적합하다. TEEN의 클러스터링 형성 기법은 LEACH와 동일하지만 데이터 전송 단계에서 (그림 4)와 같이 다른 방법을 사용한다.



(그림 4) TEEN 동작

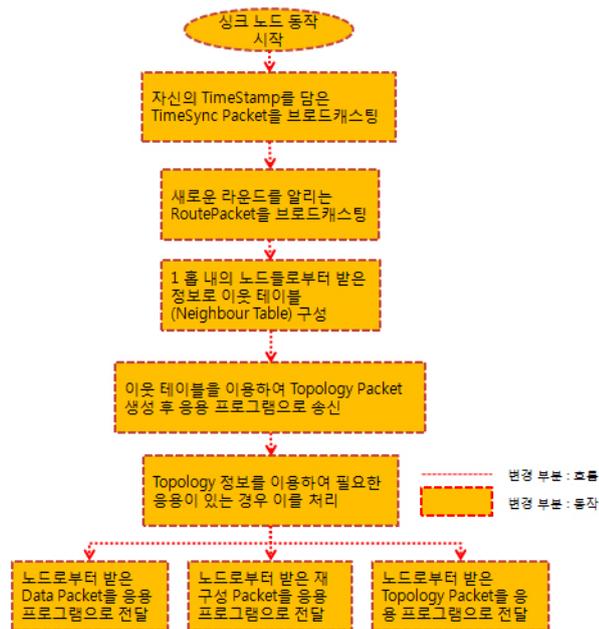
TEEN에서 센서 노드들은 클러스터 결정시간에 클러스터 헤드가 브로드캐스팅한 임계값인 H_t 와 S_t 에 기반을 두고 현재 감지된 데이터를 전송할지를 결정한다. 즉, 감지된 데이터의 값이 처음으로 H_t 를 초과하면, 이를 저장하여 해당 시간 슬롯에 전송한다. 이후에는 감지된 데이터의 값이 저장된 값보다 S_t 이상으로 큰 경우에 저장하고 해당 시간 슬롯에 전송한다.

3. 제안 프로토콜

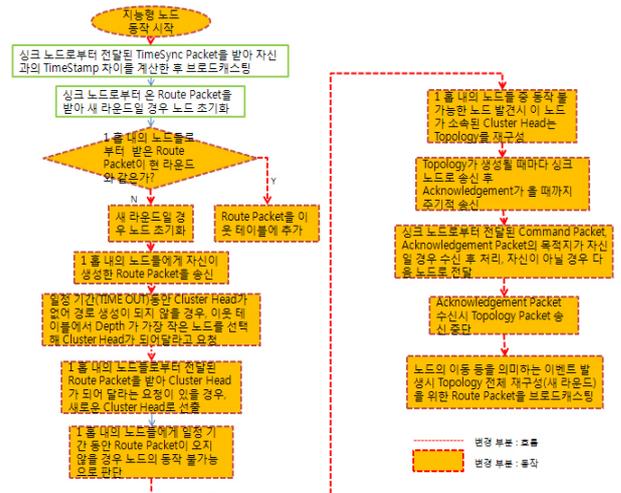
항만과 같은 환경에서는 컨테이너가 이동하기는 하지만 그 이동 정도가 미약하고 컨테이너 화물을 실시간적으로 모니터링 해야 한다는 요구사항이 있는 경우[13], 라우팅 구성의 유연이 존재하는 반응적 방식의 프로토콜보다는 사전적 방식의 LEACH 프로토콜이 보다 유리하다. 따라서 본 논문에서는 2장에서 서술한 클러스터링 기반 프로토콜인 LEACH 프로토콜에 기능을 추가하여 사용한다[15][16].

주요 변경된 내용은 통신 노드끼리 전파 범위가 닿지 않거나 음영이 생길 경우 의도적인 멀티 홉의 구성 기능과 노드의 이동 예측에 의한 네트워크 토폴로지 재구성 시간의 단축 기능이다. 기존 LEACH도 멀티 홉이 구성되지만 본 논문에서는 멀티 홉 구성의 형상을 조절하여 컨테이너 적재에 적합하도록 토폴로지를 구성하였기 때문에 급속 환경으로 둘러싸인 환경에서 발생할 가능성이 있는 음영 지역을 제거할 수 있다.

또한 노드의 움직임을 예측할 수 있는 움직임 감지 센서를 포함하여 이 센서로부터 이벤트를 발생시킴으로써 지능형 노드가 네트워크에 포함(Join)되거나 분리(Leave)될 수 있는 환경에서 빠른 속도로 그 사실을 확인할 수 있다. 이



(그림 5) 싱크 노드 동작 절차



(그림 6) 일반 노드 동작 절차

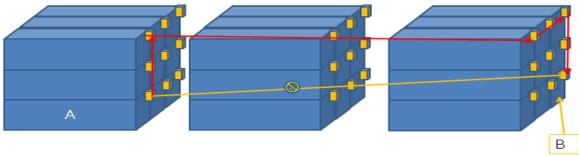
는 일반적인 LEACH 프로토콜이 사전적인 프로토콜이기 때문에 토폴로지의 변경이 미약하다라도 노드가 이동하는 환경에는 단점으로 작용할 수 있는데, 노드의 움직임 예측에 의한 토폴로지 재구성 시간을 단축함으로써 이와 같은 단점을 극복하고 항만 환경에 적합한 컨테이너 네트워크를 구성할 수 있다.

각각의 동작들은 싱크 노드의 상단에 존재할 응용 프로그램 방향으로 컨테이너 적재에 적합한 멀티 홉으로 구성된 네트워크의 토폴로지를 전달하여 토폴로지 정보가 필요할 경우 언제든지 활용할 수 있도록 한다. 이와 같은 동작을 만족시키기 위해 멀티 홉 구성 모듈, 노드 이동 예측에 의한 토폴로지 재구성 시간 단축 모듈을 구현하여 기존의 LEACH 프로토콜에 추가하였다[15][16]. 기존의 LEACH에 추가적으로 구성한 부분은 (그림 5)과 (그림 6)에 점선과 음영으로 표기하였다.

3.1 컨테이너 적재 형상에 적합한 멀티 홉 구성을 위한 알고리즘

컨테이너 적재 형상에 적합한 멀티 홉 구성을 위해서 본 논문에서는 기존의 LEACH에 몇 가지 모듈을 추가하였다. 기본적으로 멀티 홉을 구성할 수 있는 기능은 TinyOS 1.x 버전에 포함되어 있는 변형 LEACH를 활용하였고[15], 여기에 몇 가지 보완 모듈을 구현하여 라우팅 기능이 원활하도록 하였다.

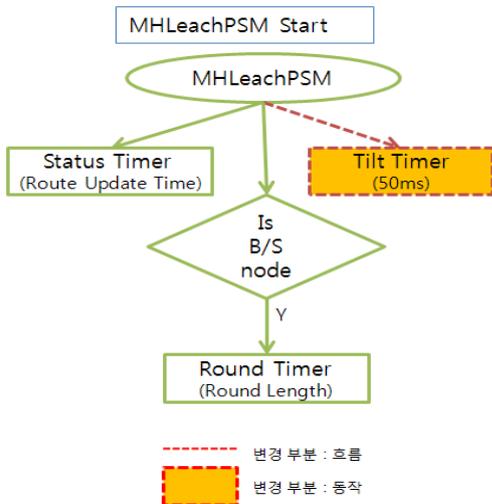
(그림 7)은 본 논문에서 컨테이너의 적재 형상에 따른 멀티 홉 토폴로지 구성의 필요성을 나타낸 것이다. 컨테이너 A가 컨테이너 B로 통신하고자 할 경우에 직접적인 전송이 불가능할 확률이 높다. 이는 급속 환경으로 구성되어 있어 전파가 제대로 전달되지 않을 가능성 및 음영이 발생할 가능성 때문이다. 이럴 경우에 적재 컨테이너의 상단 컨테이너들이 클러스터 헤드가 되고 이를 통한 트리 형태의 멀티 홉 토폴로지를 구성하면 컨테이너 적재 형상에 적합한 멀티 홉을 구성할 수 있다.



(그림 7) 컨테이너 적재 형상 및 멀티홉 구성

보통의 LEACH는 멀티 홉 구성 상황에서 클러스터 헤드의 깊이가 무작위로 결정되는데 반해 본 논문의 LEACH는 통신 지연 및 컨테이너 적재 환경을 고려하여 적정 수(5 홉 기준)의 깊이를 가지도록 설계하였다. 즉 클러스터 헤드에 Hop_Count 필드를 추가하여 이것이 1씩 감소하다가 0이 되면 그 이후의 노드는 다른 클러스터 헤드에 연결되도록 하였다. (그림 8)는 변형 LEACH의 최상단부를 구성하는 모듈이다. (그림 8)의 MHLeachPSM 모듈이 수행하는 것은 우선 각 노드를 초기화하고, 그 이후에 <표 1>의 3가지 동작을 실행하게 된다.

(그림 9)은 기존의 LEACH에서 제공되는 부분으로 주기적으로 새로운 라운드를 구성하는 부분이다. 즉 일정시간마다 전체 토폴로지를 재구성하는데 이때 클러스터 헤드



(그림 8) 변형 LEACH의 최상단 모듈

<표 1> MHLeachPSM 모듈의 내부 모듈 및 동작

모듈	동작
Status Timer	매 주기마다 Advertise Broadcast 및 노드의 Join/Leave 등의 토폴로지 변경을 감지 후 토폴로지 재구성 및 전송 이벤트 발생
Tilt Timer	진동 이벤트를 감지하기 위한 모듈
Round Timer	주기적으로 새로운 라운드를 실행, 싱크 노드에서만 실행

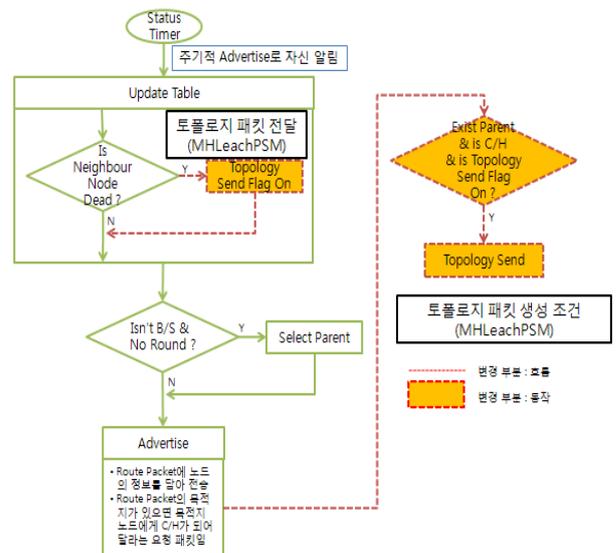


(그림 9) 라운드 타이머

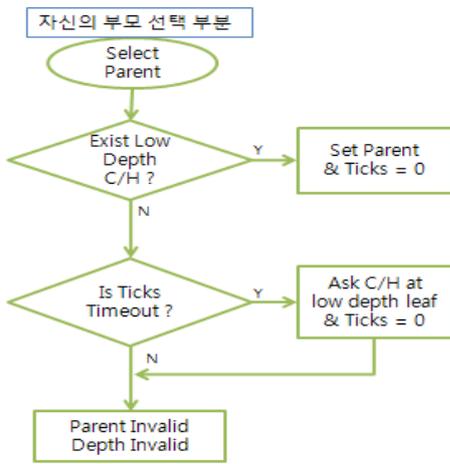
를 새로 선택하게 되며, 이는 LEACH 프로토콜의 특징이기도 하다. (그림 10)은 (그림 8)의 일부로 포함 되어 있는 Status Timer로서 매 주기마다 다음의 동작을 수행한다.

- Advertise를 이웃 노드에 브로드캐스팅 : 자신의 생존 유무, 싱크 노드로부터의 깊이, 클러스터 헤더 선출 유무, 클러스터 헤더 요구, 새 라운드 시작 유무 등을 알림
- 이웃 노드가 동작 불가능 하거나 전력이 고갈되면 노드의 추가/삭제
- 본 노드의 부모 노드가 변경될 경우, 새로운 토폴로지 패킷을 생성 후 전송

(그림 11)는 특정 노드가 부모 노드를 선택하는 알고리즘으로서 특정 노드가 토폴로지 정보 전송을 위해서 자신의 부모 노드를 선택하는 알고리즘이다. 동작은 매 Status Time 마다 Advertise(Route Packet)를 보낼 곳(부모 노드)을 탐색하는데 그 조건은 아래의 두 가지를 따른다.



(그림 10) 상태 타이머



(그림 11) 부모 노드 선택 알고리즘

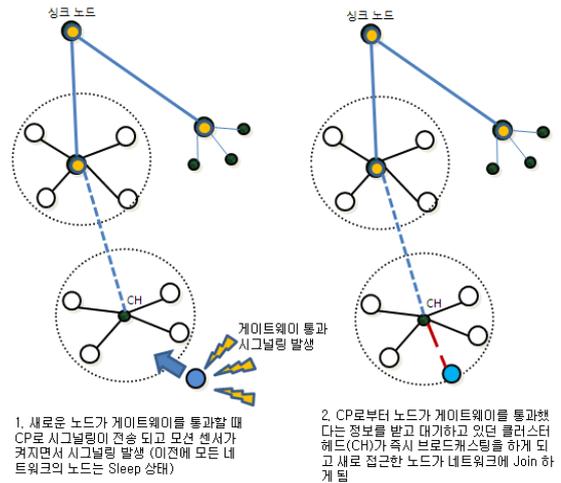
- Depth가 낮은 클러스터 헤드를 부모 노드로 선택 (5 홉을 넘을 수 없음)
- 일정시간 동안 적절한 부모 노드가 없을 시에는 Depth가 작은 리프(Leaf) 노드에게 부모 노드가 되어줄 것을 요청

이를 통해서 모든 노드들은 자신의 부모 노드를 선택할 수 있고, 네트워크 토폴로지를 결정할 수 있다. 이 알고리즘 부분으로 인해 향후 특정 지능형 노드로의 유니캐스트 명령도 전송할 수 있다.

3.2 노드 이동 예측에 의한 토폴로지 재구성 시간 단축 모듈

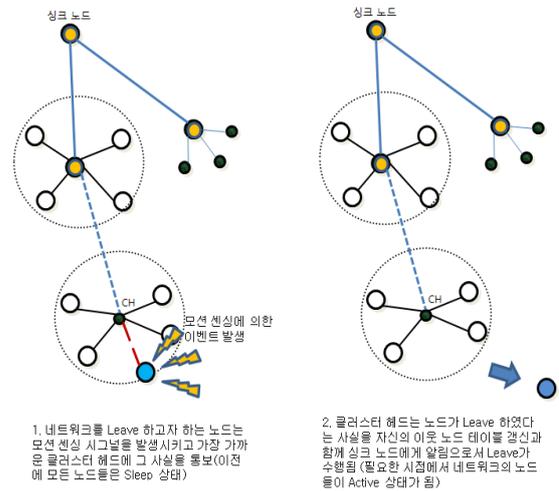
본 논문의 변형 LEACH 프로토콜의 특징 중 하나는 노드의 이동을 감지하고 이를 통해서 토폴로지 변경을 미리 예측하여 토폴로지 재구성 시간을 단축함으로써 LEACH의 사전적 동작 방식의 단점을 극복한 것이다. 즉 노드의 동작 불가능, 이동 등에 따라 특정 시간(Time Out)을 대기할 필요 없이 노드가 움직이고 있다가거나 네트워크에서 사라졌다는 이벤트를 발생시킴으로서 바로 토폴로지 재구성을 수행하게 된다. 또한 저전력을 고려하기 위해 충격 센서에 의한 네트워크에 대한 Leave and Join 기술을 도입하였다. 이 기술은 노드의 동작 정지 또는 이동으로 인해 토폴로지를 재구성하거나 또는 항상 토폴로지를 유지하기 위해서 Hello 패킷을 주기적으로 송수신할 경우 에너지의 소모가 심한 점을 개선하기 위해서 도입되었다.

아이디어는 노드가 이동 중인지 정지 상태인지를 구분하기 위한 충격 센서를 노드에 포함시켜 노드가 이동 중이라는 신호(이벤트)가 발생하면 노드에 대한 타임아웃(Time Out)을 기다릴 필요 없이 네트워크 토폴로지를 바로 재구성하도록 하는 것이다. 이러한 노드의 이동과 관련된 이벤트가 없을 경우에는 노드들이 슬립 상태에 있도록 하여 에너지를 보존할 수 있도록 하였기 때문에 결과적으로는 저전력을 달성할 수 있다. 또한



1. 새로운 노드가 게이트웨이를 통과할 때 CP로 시그널링이 전송되고 모션 센서가 깨지면서 시그널링 발생 (이전에 모든 네트워크의 노드는 Sleep 상태)
2. CP로부터 노드가 게이트웨이를 통과했다는 정보를 받고 대기하고 있던 클러스터 헤드(CH)가 즉시 브로드캐스팅을 하게 되고 새로 접근한 노드가 네트워크에 Join 하게 됨

(그림 12) 새로운 노드의 Join 동작

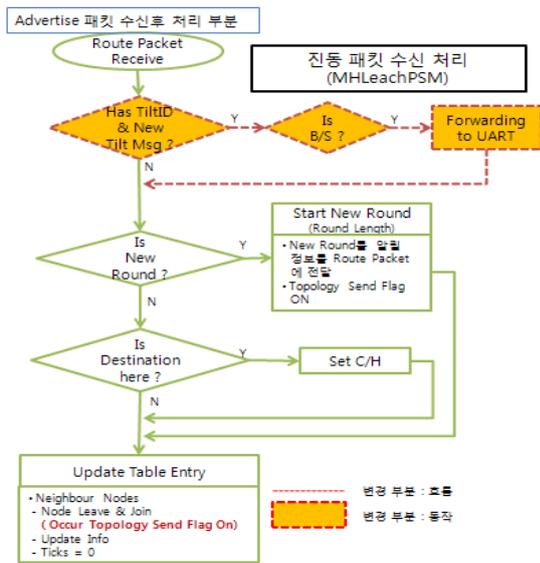


1. 네트워크를 Leave 하고자 하는 노드는 모션 센싱 시그널링 발생시키고 가장 가까운 클러스터 헤드에 그 사실을 통보(이전에 모든 노드들은 Sleep 상태)
2. 클러스터 헤드는 노드가 Leave 하였다는 사실을 자신의 이웃 노드 데이를 갱신과 함께 싱크 노드에게 알림으로서 Leave가 수행됨 (필요한 시점에서 네트워크의 노드들이 Active 상태가 됨)

(그림 13) 노드의 Leave 동작

LEACH가 사전적 방식으로서 가지고 있는 토폴로지 변화에 적절하게 대응할 수 없다는 단점 또한 보완할 수 있다. (그림 12)과 (그림 13)는 새로운 노드의 Join & Leave 동작을 보여주고 있다. 게이트웨이까지를 활용하여 LEACH 프로토콜에 Reactive적인 특성을 추가하고 있다.

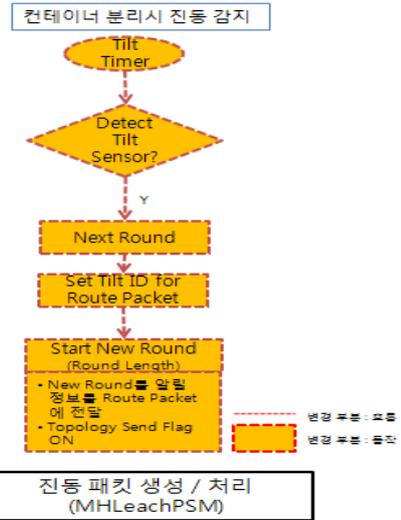
(그림 14)는 충격 센서를 활용한 Leave and Join 기술 중에서 광고된 패킷을 수신하였을 경우 처리 과정을 간략하게 플로우차트(Flow Chart)로 나타낸 것이다. 즉 이웃 노드들의 Advertise(Route Packet) 수신부로서 동작은 이웃 노드들의 Route Packet을 수신한 후 패킷의 종류와 수신자가 누구인지에 따라 몇 가지 동작을 수행한다. 우선 진동이 발생하였다는 이벤트를 수신하면 네트워크 전면 재구성을 수행한다. 이는 기존 LEACH에서 새로운 라운드로 전환하는 것과 유사한 동작이다. 네트워크 전면 재구성에 대한 패킷을 수신한 노드가 일



(그림 14) 충격 센서를 활용한 Leave & Join 처리부

반 노드가 아닌 싱크 노드일 경우에는 새로운 라운드로의 진입을 알리고 동작이 완료되면 이를 응용 프로그램으로 전달한다. 만약 전송된 패킷이 특정 노드를 목적지로 가지고 있으면 패킷 내의 becomeCluster Head를 이용해서 클러스터 헤드로 선출하게 된다. 이는 앞의 (그림 11)에 부모 노드 선택 알고리즘 동작의 일부와 연결되어 있다. 즉 특정 시간에 부모 노드 선정에 실패할 경우 가장 낮은 Depth를 가진 리프 노드를 선택하여 클러스터 헤드로 선정하는 것이다. 마지막으로 이웃 노드들로부터 받은 정보들(Alive, depth, isCluster Head, timerTicks)을 이웃 테이블에 넣는다. 이때 테이블에 이미 존재하는 노드이면 테이블에 해당 노드의 정보를 갱신하고, 존재하지 않으면 테이블에 추가한다. 이때 노드가 추가 되거나, 이미 존재하는 노드가 클러스터 헤드가 되는 등의 토폴로지 변화와 관련된 작업을 수행시 토폴로지 정보를 재구성 하여 송신하게 된다. 그리고 각 이웃 노드의 생존 여부를 확인할 때 사용되는 TIMEOUT Tick을 초기화 한다.

(그림 15)은 진동 패킷의 생성과 처리를 담당하는 부분의 간략한 플로우차트이다. 동작 순서를 간략하게 기술하면 진동 이벤트를 항상 감지하고 있는 Tilt Timer가 컨테이너 이동시의 진동 이벤트를 감지하게 된다. 진동 이벤트가 발생하고 이것이 감지되면 현재의 노드를 초기화함과 동시에 진동 이벤트가 발생 되었다는 정보를 플러딩하여 다른 노드도 초기화 시키게 된다. 즉 이와 같은 플러딩은 슬립 상태에 있는 다른 노드들에게 Active 상태가 되도록 하는 것이다. 진동의 감지와 이에 의한 모든 노드들이 Active 상태가 되면 현재 노드를 초기화함과 동시에 및 라우트 패킷에 새로운 라운드임을 알릴 정보를 주입한다. 새로운 라운드가 시작되었다는 정보를 수신하게 된 모든 노드들은 네트워크 토폴로지의 재구성을 바로 시작하게 된다.



(그림 15) 진동 패킷의 생성 및 처리에 관한 플로우차트

4. 실험 환경 및 실험 결과

실험은 네 가지 인자에 대해 수행하였다. 우선 본 논문의 주요 특성으로 실험한 인자는 네트워크의 점진적 구성 시간, 토폴로지의 재구성 시간, 컨테이너 환경에서의 통신 성공률이다. 실험을 통하여 제안한 프로토콜이 기존의 LEACH에 존재하지 않는 반응적 프로토콜 방식을 부분적으로 도입하여 노드가 움직인다는 컨테이너 네트워크의 요구 사항을 만족시켰음에도 불구하고 기존의 LEACH와 비교할 때 네트워크의 구성 시간에는 큰 차이가 없고, 토폴로지의 재구성 시간에 있어서는 반응적인 방식을 사용함으로써 오히려 단축되었음을 확인하고자 하였다. 또한 급속 환경에서 기존의 LEACH 보다는 통신 성공률이 높은 멀티 홉을 구성할 수 있다는 점을 확인하고자 한다. 다른 한 가지 인자는 노드들의 형상 유지를 위해 토폴로지 정보를 사용함으로써 발생할 수 있는 라우팅 오버헤드이다.

4.1 실험 환경

실험 환경은 <표 2>와 같다. 노드 수는 네트워크의 점진적 구성 시간에는 200개까지를 적용하였고, 네트워크 토폴

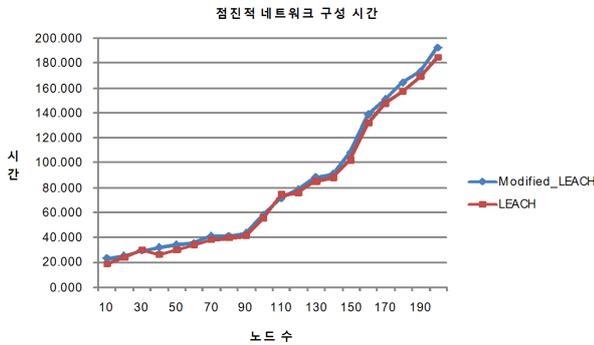
<표 2> 실험 환경

항목	값
하드웨어 플랫폼	펜티엄 4 코어2듀오 2GHz
운영체제	윈도우 XP상 cygwin에서 동작하는 TinyOS 1.15
시뮬레이터	Tinyviz
노드 수	200개(점진적 구성), 50개(네트워크 재구성 시간)
전파 범위	60m
실험 회수	점진적 구성 시간, 토폴로지 재구성 시간 (각 5회, 평균값 측정) 통신 성공률, 라우팅 오버헤드 (각 10회)

로지 재구성 시간의 측정에는 50개로 하였다. 실험 대상은 점진적 네트워크 구성 시간(초기 구성 시간), 네트워크 토폴로지 재구성 시간(전면적 재구성), 컨테이너 환경에서 통신 성공률, 라우팅 오버헤드 이다.

4.2 점진적 네트워크 구성 시간 비교

점진적 네트워크 구성 시간 실험에서는 노드 수를 증가시켜 가면서 제안 프로토콜과 기존의 LEACH 프로토콜이 네트워크를 형성하는 시간을 측정하였다. 제안 프로토콜은 반응적 프로토콜의 기능을 가지면서도 네트워크를 형성하는 시간이 기존의 LEACH와 크게 차이가 나지 않는다. (그림 16)에 점진적 네트워크를 구성하는 시간을 노드 수 200개까지 실험하였다. 평균적으로 기존의 LEACH가 네트워크 재구성 시간에 있어서는 3.047초 정도 빠르다. 그러나 기존 LEACH는 진동 센서에 의한 능동적 네트워크 구성이 불가능하다는 점이 있기 때문에 제안 프로토콜이 이러한 요구 사항 및 기능을 만족시키면서도 기존의 LEACH와 네트워크 구성 시간에서 거의 차이가 나지 않는다는 것을 (그림 16)에서 확인할 수 있다. 이와 같은 결과는 네트워크 요구 사항 중에 실시간 모니터링이 필요한 컨테이너 네트워크와 같은 환경에서 제안 프로토콜이 보다 유리할 수 있다는 것을 확인할 수 있는 하나의 인자가 될 수 있다.

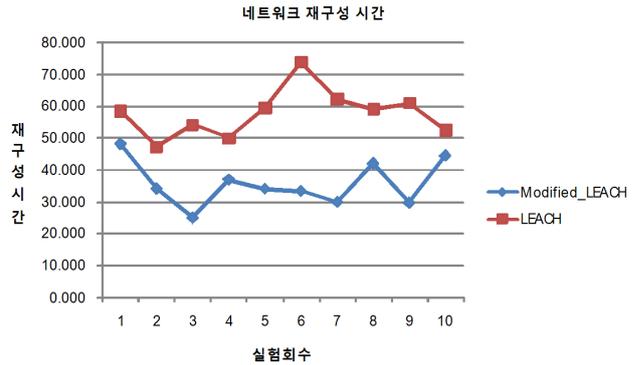


(그림 16) 점진적 네트워크 구성 시간

4.3 네트워크 토폴로지 재구성 시간 비교

네트워크 토폴로지 재구성 시간을 측정하는 실험은 노드의 움직임이 있을 경우 네트워크를 재구성하여 기능을 회복하는데 까지 걸리는 시간을 측정하는 것이다. 측정의 중심이 되는 부분은 클러스터 헤드 등의 특정 노드가 이동하여 통신이 단절되거나 지능형 노드에 장착된 진동 센서에 의해 노드가 이동함을 네트워크에 보고할 경우 제안 프로토콜이 기존 LEACH에 대해 어느 정도 빠른 네트워크 재구성을 할 수 있는가이다. 실험은 노드 수를 50개로 고정하여 실험하였다.

(그림 17)에서 확인할 수 있는 것처럼 10회의 실험 동안 제안 프로토콜이 기존의 LEACH 보다 빠른 네트워크 재구성을 하였다. 이는 기존 LEACH가 새로운 노드가 추가될



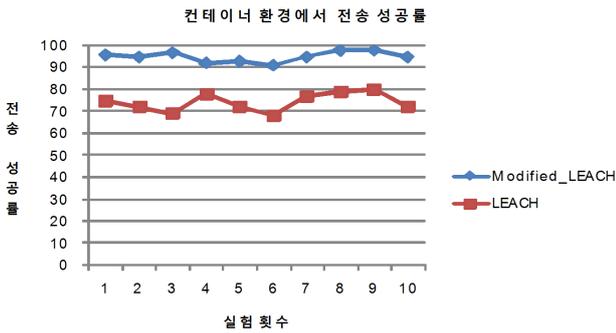
(그림 17) 네트워크 재구성 시간

때는 클러스터 헤드에 추가되지만 노드가 네트워크에서 분리되는 경우는 타임아웃을 기다리거나 라운드 갱신을 기다려야 네트워크를 재구성할 수 있기 때문이다. 이에 대해 제안 프로토콜은 평균 21.969초 정도 빠른 네트워크 재구성을 할 수 있는데 진동 센서에 의한 노드의 추가 및 분리가 능동적으로 이루어지기 때문에 더 빠른 토폴로지의 재구성이 달성될 수 있다.

컨테이너 화물에 대한 실시간 모니터링과 같은 요구 사항이 존재하는 컨테이너 네트워크에서는 반드시 충족시켜야 할 인자가 네트워크의 토폴로지가 항상 정확하게 확인되어야 한다는 점이다. (그림 17)은 그러한 요구 사항 측면에서 제안 프로토콜이 기존의 LEACH 프로토콜에 대비하여 훌륭한 성능을 나타내고 있음을 확인하고 있다.

4.4 컨테이너 환경에서 통신 성공률 비교

컨테이너 환경에서 통신 성공률 비교 실험은 기존의 LEACH에 비하여 본 논문의 제안 프로토콜이 컨테이너 환경에서 멀티 홉을 구성하는 능력과 이로 인한 통신 성공률이 어느 정도 뛰어난지를 확인하는 실험이다. 즉 통신 시야가 확보되지 못한 상태에서 기존의 LEACH가 멀티 홉을 구성하여 통신하는 방식과 본 논문에서 제안하고 있는 변형 LEACH가 구성하는 멀티 홉의 성능 비교를 하는 인자가 통신 성공률이라 할 수 있다. 노드 수 200개의 상태에서 실험하였고 패킷에 넘버를 지정하여 전송 후 그 패킷이 목적 노드에서 나타나면 성공으로, 재전송이 일어나서 시퀀스 넘버가 다른 패킷이 나타나거나 지정된 패킷이 도달하지 못하면 전송 실패로 간주하였다. 실험 1회당 패킷 50개를 목적 노드를 향하여 무작위로 전송하였다. (그림 18)는 그 실험 결과이다. 실험 결과를 확인해보면 전송 성공률의 경우 평균 20.8% 정도 제안 프로토콜이 우수한 것으로 나타난다. 이는 컨테이너 적재 방식의 형상을 유지할 수 있도록 멀티 홉을 구성하는 본 논문의 변형 LEACH가 기존의 LEACH 보다 컨테이너 환경에 적합하다는 것을 보여주고 있다. 기존 LEACH 경우 실험 결과에 반영은 되지 않았으나 통신 자체를 할 수 없는 영역도 발생하는 경우가 있었다. 즉 직각을 이루는 형태의 컨테이너 적재 야드에서는 통신 시야 확보가 중요하며, 이것이 되지 않았을 경우 다른 노드를 거



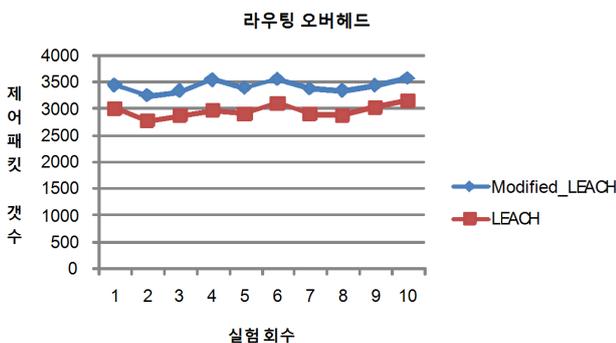
(그림 18) 통신 성공률

처 통신이 수행되도록 멀티 홉이 구성되어야 하는데 급속 환경이라는 특수성을 감안하면 토폴로지의 형태도 직각을 이루는 통신 경로를 만들어 통신하는 것이 데이터 전달을 성공시키는데 중요한 인자가 될 수 있다.

4.5 라우팅 오버헤드 비교

본 실험은 기존 LEACH와 변형 LEACH의 라우팅 오버헤드를 측정하여 제어 패킷의 수를 누적하여 그 수를 비교하였다. 제어 패킷은 주로 라우팅의 목적을 위해 생성되는 토폴로지 패킷들로서 컨테이너 적재 형상을 유지하도록 하는 토폴로지 정보의 전달과 유지를 위해서 필요하다. 앞선 실험과 마찬가지로 노드 수 200개의 상황에서 패킷 50개를 무작위로 전송하는 동안 발생하는 제어 패킷의 수를 누적하였다. (그림 19)은 그 실험 결과 이다.

라우팅 오버헤드의 경우 제안 프로토콜이 평균 462개의 제어 패킷이 더 생산되었고 이를 전체 패킷에 대한 비율로 나타내면 약 12.8% 정도 더 증가됨을 알 수 있다. 만약 노드 수가 더 증가되면 라우팅 오버헤드의 증가는 더욱 심해질 것으로 예상된다. 이는 기존 LEACH에 비해 노드들의 토폴로지를 작위적으로 형성하고 유지하기 위한 본 논문의 프로토콜 특성 때문이다. 네트워크 토폴로지의 전면적 재구성 시간 단축과 반응적 방식의 도입, 패킷 전송 성공률의 증가를 획득한 대신 라우팅 오버헤드라는 측면에서는 기존의 LEACH 보다 떨어지는 성능을 나타내고 있음을 확인하였다.



(그림 19) 라우팅 오버헤드

5. 결론

본 논문은 기존의 LEACH 프로토콜을 항만과 같은 곳에서 지능형 노드를 장착한 컨테이너가 움직이면서도 통신 환경이 급속으로 되어 있는 경우에 적합하도록 변경하는 것을 주제로 하였다. 항만은 노드 자체의 움직임은 빈번하지 않으나 분명히 움직임이 존재하고 화물의 상태를 상시 확인하여 실시간 모니터링을 수행하여야 한다는 요구 사항을 가지고 있다.

따라서 본 논문에서는 실시간 모니터링에 유리하도록 통신 지연이 적은 사전적 프로토콜인 LEACH를 선택하였다. 노드에 진동 센서를 활용하여 노드가 네트워크에 Join/Leave 하는 상황에 대해서는 반응적 네트워크처럼 동작할 수 있도록 기존의 LEACH 프로토콜을 변경시켰다.

이와 같은 변경 사항이 컨테이너 네트워크의 요구 사항을 수용하여 어느 정도의 성능을 나타내는지 확인하기 위해 네트워크의 점진적 구성 시간, 네트워크 토폴로지 재구성 시간, 통신 성공률이라는 인자를 중심으로 실험을 수행하였다. 실험 결과로서 네트워크의 점진적 구성 시간은 기존 LEACH와 거의 차이가 없으면서도 네트워크 재구성 시간에서는 더 빠른 변형 LEACH를 구현할 수 있음을 확인하였다. 또한 통신 성공률 측면에서도 급속 환경으로 구성된 컨테이너 환경에서 더 우수함을 확인하였다. 그러나 이러한 장점들을 획득하기 위해서 라우팅 오버헤드가 증가되는 것을 피할 수 없어 이러한 측면에서는 기존의 LEACH에 비해 제안 프로토콜이 단점을 가지고 있다.

본 논문에서 제안한 변형 LEACH를 활용하면 항만 같은 환경이나 실시간 모니터링이 요구되면서 노드의 움직임이 조금이라도 존재하는 환경에서는 기존의 LEACH와 비교해서 뛰어난 성능을 나타낼 수 있을 것이다.

향후 과제로는 네트워크 토폴로지 정보와 패킷의 이동 경로를 남겨 놓는 알고리즘을 추가하여 양방향 통신기능을 추가하고 이와 같은 기능을 위해서 사용되는 메모리, 네트워크 대역폭, 에너지 소모와 같은 인자도 측정하여 실험 결과에 따라 최적화시키는 것이다.

참고 문헌

[1] <http://www.etri.re.kr/> : 무선 센서 네트워크에서의 라우팅 프로토. [2] 신재욱, 권혜연, 김웅배, “이동 ad-hoc 네트워크에서 Flooding 기술”, 전자통신동향분석, 제18권, 제6호, pp.8-18, 2003. 12. [3] Dali Wei, “Clustering Ad Hoc Networks: Schemes and Classifications,” IEEE Journal of Comm, pp.920-926, 2006. 9. [4] 한옥표, 김승남, “무선 센서 네트워크를 위한 에너지 인식 하이브리드 클러스터링”, Journal of Telecommu -nication and Inforamtion, Vol.11, pp.57-64, 2007. [5] 남춘성, “무선 센서 네트워크에서 데이터 중복 방지를 위한 클러스터링 기반 라우팅 기법”, 숭실대학교 컴 퓨터공학과, 석사학위 논문, 2006. 12.

[6] 이성일, 임재성, 양성현, “에너지 효율적인 무선 센서 망 구성을 위한 Hybrid Cluster-Mesh 구성 기법”, 아주대학교 정보통신 기술연구소, 논문지 제2권, 제 1호, pp.90-94, 2007. 3.

[7] 최증식, “무선 센서네트워크 MAC계층에서 충돌 예측 및 회피를 통한 저전력 통신기법 연구”, 강원대학교, 통신멀티미디어 공학과, 석사학위논문, 2007. 2.

[8] 김영웅, 조동호, “이동 ad-hoc 망에서 트래픽 속성을 고려한 클러스터링 방안”, 한국통신학회논문지, Vol.26, No.3, pp.462-471, 2004. 8.

[9] Hans. J., “The Application-Based Clustering Concept and Requirements for InterDevice Network,” IEEE Communication Magazine, pp.108-113, 2005. 4.

[10] Ian F.Akyildiz et al., “A survey on Sensor Networks,” IEEE Communication Magazine, Vol.40, No.8, pp.102-114, Aug. 2004.

[11] C. Intangonwivat et al., “Directed diffusion for Wireless sensor Networking,” IEEE/ACM Transaction on Networking, Vol.11, No.1, pp.2-16, Feb. 2003.

[12] WenDi B. Heinzelman et al., “An Application-Specific Protocol Architecture for Wireless Microsensor Networks,” IEEE Transaction on Wireless Communication, Vol.1, No.4, pp.660-670, Oct. 2004.

[13] Juan Carlos Cano et al. “A Low Power Protocol to Broadcast Real-Time Data Traffic in a Clustered Ad Hoc Network,” IEEE Communication Magazine, Vol.2, pp.2916-2920, April, 2005.

[14] Jari Tenhunen et. al, “Stability-Based Multi-Hop Clustering Protocol,” IEEE 16th International Symposium on personal, Indoor and Mobile Radio Communication, pp.958-962, 2005.

[15] 한백전자 기술 연구소, “유비쿼터스 센서 네트워크 시스템”, Info-Tech Corea, 2007. 11. 4

[16] 남상엽, 송병훈, “MOTE-KIT를 이용한 무선 센서 네트워크 활용”, 상학당, 2006. 2. 25



이 양 민

e-mail : manson23@nate.com
 2000년 동아대학교 컴퓨터공학과(공학사)
 2002년 동아대학교 대학원 컴퓨터공학과 (공학석사)
 2006년 동아대학교 대학원 컴퓨터공학과 (공학박사)

2007년 BK21 고성능 멀티미디어 사업팀 연구원
 2007년~현재 동아대학교 컴퓨터공학과 (시간강사)
 관심분야 : 유비쿼터스 컴퓨팅, Ad-hoc 네트워크 등



이 기 원

e-mail : koyrim@nate.com
 2002년~현재 동아대학교 컴퓨터공학과 학사과정
 관심분야 : 유비쿼터스 네트워크, WPAN 등



곽 광 훈

e-mail : paxpia@dau.ac.kr
 1989년 부산대학교 전기공학과(공학사)
 1999년 포항공과대학교 정보통신공학과 (공학석사)
 1989년~1999년 포스코, 포스테이타
 2000년~2002년 i2soft 연구소장

2002년~2003년 한국전자통신연구원 선임연구원
 2005년~현재 동아대학교 미디어디바이스연구센터 책임연구원
 관심분야 : WPAN, IPv6, 유비쿼터스 네트워크, 퓨처인터넷



이 재 기

e-mail : jklee@dau.ac.kr
 1978년 영남대학교 전자공학과(공학사)
 1983년 영남대학교 대학원 전자공학과 (공학석사)
 1990년 일본 동경대학 전자정보공학전공 (공학박사)

1984년~1990년 한국전자통신연구소 연구원
 1990년~현재 동아대학교 컴퓨터공학과 교수
 2000년~2001년 일본 마쿠하리기가비트연구센터 초빙 연구원
 관심분야 : 차세대 네트워크, 유비쿼터스 컴퓨팅, 분산시스템 등