

RFID 기반 물류창고 시스템을 위한 센서 네트워크 구축

(Sensor Network Deployment for Warehouse Management System based on RFID)

이 신 형 [†] 이 치 영 [†] 김 동 신 [†] 이 찬 행 [†]
 (ShinHyoung Lee) (ChiYung Lee) (DongShin Kim) (ChanHaeng Lee)

이 원 준 ^{**} 민 성 기 ^{***} 유 혁 ^{**}
 (WonJun Lee) (SungGi Min) (Hyuck Yoo)

요 약 물류창고는 수시로 많은 물류가 입고/출고됨에 따라 수시로 재고 현황이 변화하고 이를 빠르고 정확하게 관리하는 것은 물류 관리에 있어서 매우 중요하다. 이를 위해 RFID 기반의 센서 네트워크를 구축하여 기존의 수작업으로 하던 작업을 빠르게 수행할 수 있도록 한다. 센서 네트워크를 구축하기 위해 에너지 기반으로 클러스터링을 하고 이를 기반으로 계층적 라우팅을 통해 데이터 전송을 가능하게 하였다. 또한 클러스터간 이동성을 지원하여 각 노드가 이동하여도 연결성이 보장되도록 하였다. 이렇게 구축된 네트워크는 실제 RFID 리더와 함께 연동되어 물류창고에서 사용되었다.

키워드 : RFID, 센서네트워크, 클러스터, 이동성, 주소지정, IPv6, 라우팅, 물류창고

Abstract Warehouse has many changes for the stocks by entering and taking. Management about the stocks is important. To manage, sensor network based on RFID is deployed in warehouse. Sensor network uses cluster based on energy and hierarchical routing. Supporting mobility between clusters makes guarantee connectivity for mobile nodes.

Key words : RFID, Sensor Network, Cluster, Mobility, Addressing, IPv6, Routing, Warehouse

1. 서 론

RFID는 Radio Frequency Identification의 약자로 많은 응용분야를 지니고 있다. RFID의 특징은 RFID 리더에서 비접촉식으로 RFID tag의 정보를 읽어올 수 있으므로 해서 Line of Sight가 보장되지 않는 환경에서도 원하는 정보를 정확히 가져 올 수 있다[1].

이러한 RFID 기술이 가장 크게 활용될 수 있는 분야는 물류이다. 기존의 물류 시스템에서는 입고, 출고 및 재고정리시에 모든 물류를 사람이 직접 손으로 계산함으로써 현황을 파악함으로써 인해 시간이 많이 소요되었다. 반면 RFID를 이용한 물류 시스템에서는 이용하여 물류의 이동 및 재고 현황을 실시간으로 확인하고 그에 따른 정책을 빠르게 결정할 수 있다. 특히 물류창고 시스템에서 가장 많은 시간을 차지하는 재고 파악을 빠르게 수행함으로써 인력을 최소화하고 효율적으로 정책을 결정할 수 있다[2].

RFID 물류시스템을 위해서는 RFID Tag와 Reader,

· 본 연구는 산업자원부의 성장동력 사업 지원으로 수행되었음(과제번호 10016756)

[†] 학생회원 : 고려대학교 컴퓨터학과
 shlee@os.korea.ac.kr
 cylee@os.korea.ac.kr
 dongshin@korea.ac.kr

^{**} 종신회원 : 고려대학교 컴퓨터학과 교수
 wlee@korea.ac.kr
 hxy@os.korea.ac.kr

^{***} 정 회 원 : 고려대학교 컴퓨터학과 교수
 sgmin@korea.ac.kr

논문접수 : 2007년 9월 6일

심사완료 : 2008년 1월 8일

Copyright©2008 한국정보과학회 : 개인 목적이나 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 받고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제1호(2008.2)

그리고 서버가 필요하다. 또한 리더와 서버의 통신을 위한 네트워크 구축이 필요하다. 사람이나 지게차에 장착되어 있는 리더는 RFID Tag를 읽고 그와 함께 네트워크를 통해 서버에게 데이터를 전달하여야 한다. 따라서 에너지의 효율적인 사용이 매우 중요하다. 또한 사람이 가지고 다니는 리더의 크기는 많은 제한을 갖는다. 리더에 다양한 센서를 장착함으로써 리더 주위의 환경 정보를 수집할 수 있고, 이는 다양한 물류의 특성에 맞도록 환경을 조절할 수 있도록 한다. 따라서 에너지 효율적이고 단말기의 자원을 많이 사용하지 않는 센서네트워크 프로토콜이 적합하다.

본 논문에서는 RFID기반 물류창고 시스템을 위한 네트워크를 제안하고 이를 실제로 구축한 내용을 설명한다.

본 논문은 다음과 같이 구성되어 있다. 1장의 서론에 이어 2장에서는 전체 시스템의 모습을 설명하고, 3장에서는 기본이 되는 클러스터 기법, 4장에서는 IPv6[3] 기반의 주소 지정 기법 및 클러스터간 이동 기법에 대해 설명한다. 5장에서는 클러스터를 이용한 계층적 라우팅 기법에 대해 설명하고 6장에서 결론을 통해 마무리 짓겠다.

2. 시스템 구성

물류창고를 위한 센서네트워크 프로토콜은 리더와 연결된 단말기에서 실행된다. 리더가 읽은 태그 정보는 단말기에 저장되고 단말기는 센서네트워크를 통해 중앙의 서버에게 정보를 저장한다.

네트워크가 초기화 되면 클러스터 알고리즘에 의해 클러스터 헤더 및 노드가 정해지고 클러스터에 따라 주소가 정해진다. 주소가 정해지면 라우팅 프로토콜이 실행되어 경로가 설정되고 단말간 통신이 가능해진다.

각 부분의 메시지 흐름은 먼저 클러스터 헤더 선출 알고리즘에 의해 헤더가 선출되고 멤버 노드가 정해지면 각 클러스터 구성 서비스는 메시지를 통해 주소를 지정하도록 한다. 주소 지정 서비스가 주소 지정 알고리즘에 의해 주소를 지정하고, 지정된 주소를 라우팅 프로토콜에게 메시지를 통해 넘겨준다. 그리고, 중앙 서버로 자신의 주소를 전송한다. 이때 중앙 서버까지의 경로를 알지 못하기 때문에 라우팅 프로토콜이 동작하여 중앙 서버까지의 경로를 찾는다. 중앙 서버까지의 경로를 구



그림 1 시스템 전체 구성도

한 후에는 주소 업데이트 작업이 완료된다.

네트워크 구축 환경은 윈도우XP 기반으로 개발하였고 사용한 MAC은 802.11[4]을 사용하였다.

3. 클러스터 구성 기법

기존의 클러스터 구성 기법으로는 노드 클러스터 헤더 적합도를 통해 클러스터를 구성하는 Distributed Clustering Algorithm(DCA)[5], 클러스터 반경 내에서 가장 낮은 식별자가 클러스터 헤더가 되는 Linked Cluster Algorithm(LCA)[6], 그리고 노드에 일정한 가중치를 두어 클러스터를 구성하는 Weighted Clustering Algorithm(WCA)[7] 등이 있다. 본 논문에서 사용되는 클러스터 구성 기법은 서로의 통신 및 연산을 최소화하여 클러스터를 구성한다.

클러스터는 처음 네트워크가 초기화 될 때, 그리고 클러스터 헤더의 에너지가 정해진 양 이하로 떨어질 때 구성 및 재구성된다.

3.1 동작 방식

모든 노드는 클러스터 헤더가 주기적으로 송신하는 비콘 메시지를 일정시간동안 기다린다. 일정 시간 비콘을 받지 못한 노드중 일정한 에너지 이상 갖고 있는 노드는 임의로 헤더가 된다. 헤더가 된 노드는 4장에서 설명하는 주소 생성 알고리즘에 의해 자신의 주소를 생성하고 즉시 클러스터 헤더 비콘을 송신한다. 비콘을 받은 노드는 클러스터 멤버로써 동작한다.

클러스터 헤더는 에너지가 일정 수준 이하로 떨어지기 전까지 계속 헤더로 동작한다. 클러스터 헤더의 에너지가 일정 수준 이하로 떨어지면 클러스터 헤더로써의 동작을 멈추고 더 이상 비콘을 송신하지 않는다. 일정 시간동안 비콘을 받지 못한 주변의 멤버 노드들은 다시 헤더 선출 알고리즘에 의해 헤더를 선출한다. 전체 클러스터 구성 기법에 대한 순서도는 그림 2와 같다.

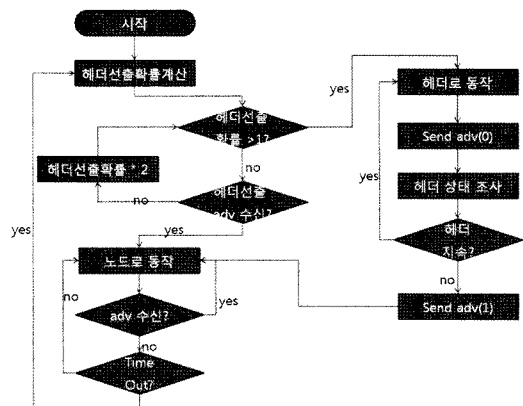


그림 2 클러스터 구성 기법의 순서도

3.2 클러스터 헤더 선출 알고리즘

무선 리더의 에너지를 고려하여 클러스터 헤더 선출 메커니즘을 설계한다. 클러스터의 반경이 너무 넓으면 클러스터 헤더는 통신에 많은 에너지를 소모하므로 1-hop으로 클러스터의 범위를 한정한다. 1-hop 반경의 클러스터를 생성할 때 필요한 사항들을 설계한다. 헤더로 선출된 무선 리더는 트래픽이 자신에게 몰리게 되므로 다른 리더보다 더 많은 에너지를 소비하게 된다. 무선 리더의 에너지는 한정되어 있으므로 헤더 리더의 에너지 소모를 최소화 하여 클러스터의 재구성을 최소화한다. 따라서 무선 리더기의 에너지 잔량에 따라 클러스터 헤더를 선출하는 기법을 설계해야 한다. 또한 클러스터 헤더를 동적으로 변경하여 모든 노드가 균등하게 에너지를 소비하는 메커니즘이 필요하다.

클러스터 헤더 선출 시 고려할 사항은 다음과 같다.

- 클러스터 헤더의 에너지 잔량 :
클러스터 헤더는 멤버 노드로부터 끊임없이 데이터를 받고 이를 통합하는 역할을 한다. 또한 노드들이 서로 충돌하지 않도록 스케줄링을 관리하는 기능을 수행해야 하므로 많은 에너지의 소모가 예상된다. 이러한 에너지 소모를 전체 네트워크에서 균형적으로 소비하기 위해서 가능한 잔량이 많은 노드를 클러스터 헤더로 선출해야 하겠다.
- Cover할 수 있는 member들의 수 및 분포 :
클러스터 헤더가 지나치게 많은 노드를 관리하게 되면 노드들이 데이터를 보낼 수 있는 슬롯이 줄어들게 되고 따라서 데이터를 원활히 보낼 수 없는 상황이 발생할 수 있다. 또한 노드의 개수가 많아질수록 헤더가 데이터를 받아야 되는 시간과 양이 늘어나게 되므로 에너지의 소모가 많아지게 된다. 따라서 전체 네트워크를 원활히 운영하기 위해서 클러스터 헤더간 노드의 개수를 조정하는 메커니즘이 필요하다.

- 부하 정도 :
물류 시스템의 특성상 특정 위치에 있는 멤버 노드는 특별히 이벤트가 많이 발생하여 데이터를 많이 보내는 현상이 나타날 수 있다. 이러한 지역의 클러스터는 클러스터 헤더가 범위를 줄여서 클러스터 내에서 데이터가 일정량 이하로 교환하도록 헤더 선출에서 고려해야 한다.

헤더 선출 알고리즘이 동작하면 먼저 전체 네트워크 상에 클러스터 헤더의 비율을 결정한다. 너무 작은 에너지를 갖는 헤더가 선출되는 것을 막기 위해 클러스터 헤더가 갖는 최소 에너지 값을 정한다. 클러스터 헤더가 될 수 있는 후보 노드들은 최소 에너지값 이상을 갖는 노드들로 이루어지고, 에너지값이 높을수록 헤더가 될 확률이 높아진다.

클러스터 헤더 선출 확률은 클러스터 헤더 비율에 따라 0에서 1 사이의 값으로 정해진다. 클러스터 헤더 선출 확률이 1이 되면 클러스터 헤더로 선출되어 비콘을 송신한다. 클러스터 헤더 선출 확률은 일정한 시간 동안 기다려서 주위에 다른 클러스터로부터의 비콘이 없으면 기존의 클러스터 헤더 선출 확률*2를 하여 1과 비교하여 작은수를 선택한다. 그 결과 나온 값을 새로운 클러스터 헤더 선출 확률로 정한다. 그리고 이 동작을 클러스터 멤버가 되거나 클러스터 헤더가 되기 전까지 반복한다.

헤더 선출 알고리즘에 대한 수도코드는 다음 그림 3과 같다.

```

클러스터 헤더 비율 입력
클러스터 헤더 최소 에너지 입력
If 현재 에너지 > 클러스터 헤더 최소 에너지 then
  클러스터 헤더 선출 확률 결정 (0~1)
  While(1)
    If 클러스터 헤더 선출 확률 = 1 then
      클러스터 헤더 선출
      클러스터 헤더 비콘 송신
      Return
    Else
      If 일정시간동안 클러스터 헤더 비콘 수신
then
      클러스터 멤버로 설정
      Return
    Else
      클러스터 헤더 선출 확률
      = min(1, 클러스터 헤더 선출 확률 * 2)
    Endif
  Endif
End While
Else
  클러스터 멤버로 설정
  클러스터 헤더 비콘 송신

```

그림 3 헤더 선출 기법 알고리즘의 수도코드

본 기법에 의하면 클러스터 헤더선출 기법이 간단하기 때문에 헤더 선출에 따른 오버헤드가 적고 주변과의 데이터 교환이 필요 없다. 헤더 선출 과정에서 주변 노드와 어떠한 정보 교환도 없으므로 에너지를 많이 절약할 수 있다.

4. IPv6(3) 기반의 주소 지정 기법 및 클러스터간 이동 기법

4.1 IPv6 기반의 주소 지정 기법

이동성 지원을 위해서는 전체 네트워크에서 노드를 구분할 수 있는 고유한 ID와 현재 위치한 네트워크에 따른 네트워크 ID가 필요하다. RFID 시스템에서 RFID 리더의 ID는 리더를 구분하는 고유한 식별자이다. 따라

서 클러스터환경에서 리더의 ID를 사용하여 고유한 노드 ID를 가질 수 있게 된다.

노드가 현재 속한 클러스터에서 부여 받은 임시 주소를 클러스터 헤더의 정보와 클러스터 노드 자체 정보를 혼합하여 생성한다. 클러스터 헤더도 클러스터 노드와 마찬가지로 RFID 리더와 연결되므로, 고유한 ID를 갖고 있다. 클러스터 노드는 클러스터 헤더의 ID에 노드의 고유 ID를 더하여, 클러스터에서의 네트워크 주소를 생성하게 된다.

네트워크 주소를 위한 주소 체계는 IPv4나 IPv6가 사용될 수 있다. 임시 주소는 클러스터 헤더의 ID와 노드의 ID를 더하여 구성하게 되므로, IPv4나 IPv6 주소 중 일부 비트를 두 ID에 할당하여 사용하게 된다. IPv6의 경우는 128비트의 길이를 갖고 있기 때문에, ID를 그대로 사용할 수 있는 반면, IPv4의 경우는 32비트에 ID를 넣기 위하여 매핑 테이블을 추가로 구성하여야 가능하다. 따라서 네트워크 주소는 IPv6 기반의 주소체계를 이용한다. IPv6는 64비트의 프리픽스와 64비트의 포스트픽스로 구성된다. 프리픽스는 클러스터 주소를 나타내고 포스트픽스는 노드 주소를 나타낸다. 클러스터 헤더는 RFID 리더의 시리얼을 통해 클러스터 주소를 생성하고 포스트픽스는 1로 지정된다. 클러스터 멤버 노드는 클러스터 주소를 프리픽스로 갖고 자신의 RFID 리더 시리얼을 통해 노드 주소를 생성한다.

4.2 이동성 지원을 위한 위치 관리 기법을 위한 구성

기존의 위치 관리 기법으로는 MobileIPv6[8][9]가 대표적이다. 본 논문에서 사용하는 이동성 지원을 위한 위치 관리 기법은 특별한 에이전트가 없이 클러스터 헤더 및 서버와의 통신을 통해 이동성을 지원하고 위치를 관리한다.

위치 관리를 위한 주요 구성 요소로는 위치 관리 서버, 클러스터 헤더, 클러스터 노드가 있다.

- 위치관리 서버 :
위치 관리 서버는 각 노드의 고유 ID와 현재 클러스터에서 부여 받은 임시 주소를 관리해주는 서버이다. 이러한 위치 관리 테이블은 노드가 새로운 클러스터에 이동할 때마다 갱신되며, 노드와 통신하고자 하는 노드에게 정보가 제공된다.
- 클러스터 헤더 :
클러스터 헤더는 자신의 고유 ID를 노드에 전달하여 클러스터 노드가 클러스터 헤더의 ID를 포함하는 클러스터 임시 주소를 생성하도록 한다. 클러스터 헤더는 자신의 ID 정보를 주기적으로 브로드캐스트 한다. 또한, 모든 클러스터 노드는 클러스터 헤더를 통하여 서버로 패킷을 전달하도록 된다.
따라서 위치 관리에 관련된 메시지 역시 클러스터 헤

더에서 서버로 전달하는 기능을 수행한다.

- 클러스터 노드 :
클러스터 노드는 클러스터 헤더에서 전달되는 헤더 ID를 받아 클러스터 영역을 인식하고, 새로운 영역으로 이동한 경우 헤더 ID와 자신의 고유 ID를 사용하여 주소를 생성한다. 그리고 새로운 주소를 서버에 알리는 위치 갱신 절차를 거친다.

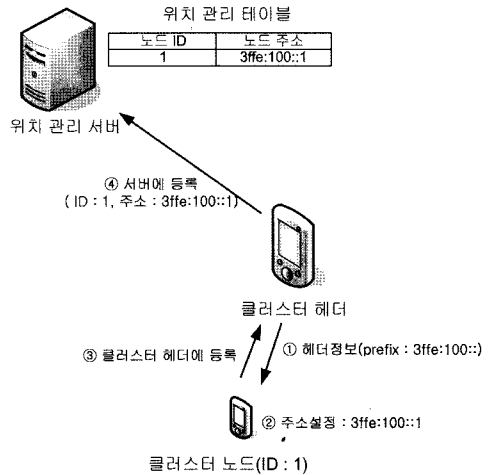


그림 4 IPv6를 사용한 위치 관리 절차

그림 4는 클러스터 노드, 클러스터 헤더, 위치 관리 서버가 IPv6를 사용하여 위치관리를 하는 절차를 나타내고 있다. 클러스터 헤더는 자신의 헤더 정보를 브로드캐스트 한다. 클러스터 노드는 헤더에서 보낸 정보를 사용하여 자신의 ID와 합하여 노드의 임시 주소를 생성하고, 그것을 클러스터 헤더에 전달한다. 클러스터 헤더는 클러스터 노드를 대신하여 위치 관리 서버에 노드의 정보를 갱신한다.

그림 5, 그림 6, 그림 7은 각각 위치 관리 서버, 클러스터 헤더, 클러스터 노드에서의 처리 알고리즘을 나타내고 있다.

위치 관리 서버에서는 클러스터 노드의 ID와 노드의 현재 주소를 갖는 addr_table이라는 전역 테이블을 유지한다. 이 테이블의 각 엔트리는 유효시간(lifetime)을 갖고 있다. 만약 이 시간이 0이 되면, 엔트리는 유효하지 않은 것으로 간주하여 addr_table에서 지워지게 된다. 이것은 클러스터 노드가 오동작에 의하여 사라진 경우 잘못된 정보가 계속 남아있는 것을 방지하기 위한 것이다. 위치 갱신 메시지를 수신하면, addr_table에서 ID를 검색하게 되고, 만약 존재하지 않으면 새로운 엔트리를 추가하게 된다. 그리고 클러스터 노드가 정상적으로 동작하고 있으므로 유효시간을 갱신하여 엔트리가

```

전역 변수:
  addr_table(nodeID, address, lifetime)
  //클러스터 노드 ID 와 주소의 매핑 테이블

If 위치 갱신 메시지수신 then
  ID = 클러스터 노드의 ID
  new_addr = 새로운 주소
  if addr_table 에 ID 가 없으면 then
    addr_table 에 ID 추가
  endif

  addr_table 의 클러스터 노드 ID 에 new_addr 설정
  addr_table entry 의 lifetime 갱신

  클러스터 헤더로 위치 갱신 Ack 메시지 전송
Endif

If addr_table entry 의 lifetime == 0 then
  addr_table 의 entry 를 제거
Endif
    
```

그림 5 이동성 지원 알고리즘(위치관리 서버)

```

주기적으로 :
  클러스터 헤더 ID 메시지 전송

If 위치 등록 메시지 수신 then
  서버에 위치갱신 메시지 전송
Endif

if 위치 갱신 Ack 메시지 수신 then
  클러스터 노드에 에 위치 등록 Ack 메시지 전송
Endif
    
```

그림 6 이동성 관리 알고리즘(클러스터 헤더)

```

전역 변수 :
  my_addr : 현재 주소
  updateTime: 갱신주기

If 클러스터 헤더 ID 정보 메시지 수신 then
  new_addr = 헤더의 ID 정보 + 노드의 고유 ID;

  if my_addr != new_addr 또는 갱신주기 then
    my_addr = new_addr
    interface 에 주소 설정
    cluster header 에 위치 등록 메시지 전송
  Endif
Endif
    
```

그림 7 이동성 지원 알고리즘(클러스터 노드)

지워지는 것을 방지한다. 이와 같은 절차가 모두 정상적으로 수행되면 ack 메시지를 클러스터 헤더로 전송한다.

클러스터 헤더는 클러스터 헤더의 존재를 알리는 메시지를 주기적으로 전송한다. 클러스터 노드는 이 메시지를 수신하여 클러스터 영역 정보를 얻고 그에 따른 핸드오프를 하게 된다. 위치등록 메시지를 클러스터 노드로부터 수신하게 되면, 서버에 위치 갱신 메시지를 전송하게 된다. 클러스터 노드가 직접 서버로 위치 갱신 메시지를 전송하지 않는 이유는 모든 클러스터 노드의 패킷은 클러스터 헤더를 통하여 전달되기 때문이다.

클러스터 노드는 자신이 현재 속한 클러스터 영역을 클러스터 헤더로부터 수신한 클러스터 헤더 ID 정보를 통하여 인식하고, 새로운 클러스터 영역으로 이동한 것을 감지하면, 위치 갱신 절차를 수행하게 된다. 새로운 클러스터 영역을 감지하지 않더라도, 위치 관리 서버에서 유효시간에 의하여 정보가 삭제되는 것을 방지하기 위하여 일정한 갱신 주기가 되면 새로운 영역에 이동한 것과 마찬가지로 위치 갱신 절차를 수행한다.

4.3 클러스터 노드의 위치 정보 제공

클러스터 노드가 새로운 클러스터로 이동하는 경우 갱신되는 위치 관리 테이블은 노드간 통신 또는 RFID 서버와 노드간 통신에서 사용된다. 그림 8은 노드간 통신 시에 위치 정보 제공 절차를 나타내고 있다. 노드 A는 노드 B와 통신하려 한다. 이 경우 노드 A는 노드 B의 고유 ID인 2를 알고 있다. 노드 A는 노드 B의 현재 주소를 얻기 위하여 위치 관리 서버에 요청을 하게 된다. 클러스터링 환경하기 때문에 노드의 모든 전송은 클러스터 헤더를 통과하게 된다. 클러스터 헤더는 노드의 요청을 위치 관리 서버에 전달하고, 위치 관리 서버는 노드 ID 2에 대한 주소값인 3ffe:101::2를 노드 A에 전달한다. 노드 A는 노드 B의 현재 주소를 알고 있으므로 노드 B와 통신하여 데이터를 주고받을 수 있게 된다.

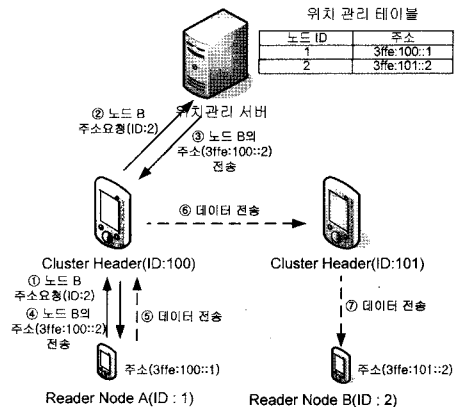


그림 8 노드간 통신을 위한 위치정보 제공 절차

5. 클러스터를 이용한 계층적 라우팅 프로토콜

5.1 클러스터를 이용한 계층적 라우팅 기법 [10]

무선 ad-hoc 네트워크에는 proactive 라우팅 프로토콜과 reactive 라우팅 프로토콜이 있다. Proactive 라우팅 프로토콜에는 OLSR(Optimized Link State Routing Protocol)[11], DSDV(Destination Sequenced Distance Vector)[12] 등이 있고, reactive 라우팅 프로토콜에는 AODV(Ad-hoc On-demand Distance Vector)[13,14], DSR(Dynamic Source Routing)[15] 등이 있다. 기존의 무선 ad-hoc 라우팅 프로토콜은 경로 탐색 및 라우팅 정보 유지를 위해 기본적으로 브로드캐스트 전송을 필요로 한다. 그러나 브로드캐스트는 네트워크를 구성하는 모든 노드에게 메시지를 전송함으로써 큰 오버헤드를 갖는다. 이런 이유로 네트워크 전체에 걸리는 오버헤드를 줄이기 위해 클러스터링 기법을 사용한다. 클러스터에 기반의 대표적인 라우팅 프로토콜로는 CBRP(Cluster Based Routing Protocol)[16]가 있다. 본 논문에서는 그림 9와 같이 클러스터 기반의 계층적 라우팅 기법을 사용한다. CBRP는 기본적인 경로 구성은 동일하지만 CBRP에서 제공하는 클러스터 멤버간 통신은 지원하지 않는다. 이는 물류창고 시스템에서 모든 통신은 클러스터 멤버와 서버 사이에서 이루어지는 특성 때문이다[17].

클러스터 헤더와 클러스터 멤버 사이에 멀티 홉으로 구성할 경우, 헤더와 멤버 노드 사이에도 라우팅 프로토콜이 필요하게 된다. 그러므로 3장에서 언급한 바와 같이 클러스터와 멤버 노드는 1 홉으로 통신이 가능하도록 한다. 이로 인해 헤더와 멤버 사이에 라우팅 프로토콜이 불필요하게 된다. 헤더와 멤버가 1 홉으로 통신이 가능하기 때문에 헤더와 헤더 간의 통신은 최대 3 홉으로 가능하다. 즉, 헤더와 헤더 간에는 통신을 위한 라우팅 프로토콜이 필요하다. 만약 헤더가 상대적으로 장거리의 통신이 가능한 MAC을 사용한다면 인접 클러스터 헤더와 1 홉으로 통신이 가능하다. 하지만 이는 클러스터 헤더가 될 수 있는 후보 노드의 수를 줄이는 결과를 초래하여, 클러스터 구성의 유연성이 떨어지는 단점이 있다. 반드시 장거리 통신이 가능한 MAC을 사용하는 노드만이 헤더가 될 수 있기 때문이다. 따라서 라우팅 프로토콜의 사용으로 인한 오버헤드를 줄이면서 사용하는 MAC의 종류에 관계없이 유연하게 클러스터를 구성할 수 있도록 헤더와 멤버 간에는 1 홉, 헤더와 헤더 간에는 최대 3 홉이 되도록 한다.

구성한 클러스터 구조는 그림 9와 같이 상·하위 계층으로 나뉜다. 각 클러스터 헤더는 인접 클러스터 헤더와 통신한다. 앞서 말한 바와 같이, 클러스터 헤더와 클러스터 멤버는 1 홉으로 통신을 하는 반면, 헤더 간의

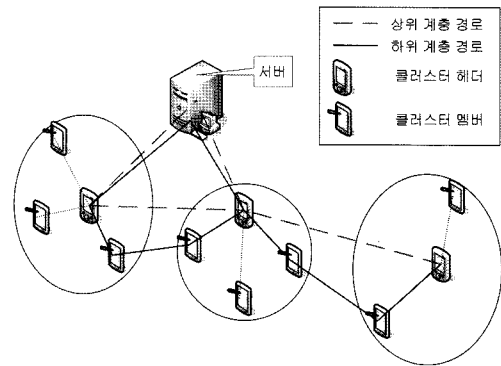


그림 9 계층적 클러스터 기반의 라우팅 경로

통신은 최대 3 홉으로 가능하다. 그러므로 클러스터의 하위 계층에서는 인접 클러스터 헤더와 통신하기 위해 메시지의 TTL 값을 3으로 설정하여 전체 네트워크에 브로드캐스트 되는 것을 막을 수 있다. 하위 계층에서 만들어진 라우팅 경로는 실제로 클러스터 헤더 사이에 3 홉이지만, 클러스터의 상위 계층에서는 1 홉의 라우팅 경로로 간주한다. 따라서 상위 계층의 라우팅 경로는 그림 1에 나타난 형태로 구성된다. 이로써 하위 계층에서 만든 경로에 포함되지 않는 노드들은 통신하지 않도록 하여 전체 네트워크의 오버헤드를 줄일 수 있다[4].

또한, 라우팅 경로를 유지하기 위해서는 주기적인 컨트롤 메시지 전송이 요구된다. 그러나 이는 네트워크에 컨트롤 메시지 전송으로 인한 오버헤드를 발생시킨다. 이 오버헤드를 줄이기 위해 on-demand 방식으로 동작하는 AODV[13,14] 라우팅 프로토콜을 기반으로 한다. 즉, 데이터를 보내질 때에만 컨트롤 메시지를 전송하여 라우팅 경로를 생성하고 유지하여 네트워크의 효율성을 향상시키고자 하였다.

5.2 라우팅 프로토콜의 구현 [18]

라우팅 프로토콜은 윈도우 플랫폼 기반의 유저 레벨의 응용 프로그램 형태로 구현하였다. 이를 위해 응용 계층에서 오버레이 네트워크를 구성하여 이용한다. 구성된 네트워크에서 프로토콜과 커널의 관계는 그림 10과 같다.

구현된 프로토콜은 인접 클러스터 테이블과 이웃 노드 정보를 가진 테이블을 유지하며, 두 테이블의 정보를 이용하여 윈도우의 라우팅 테이블을 수정하는 방식으로 동작한다.

프로토콜의 전체 프로세스는 그림 11과 같이 3 가지로 구분할 수 있다.

- 데이터 구조체 및 장치 초기화
- 경로 탐색(path discovery) 루틴
- 이벤트(event) 처리 루틴

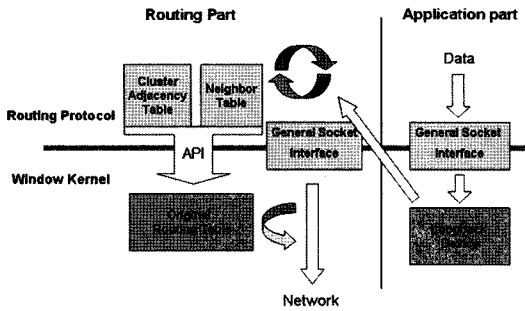


그림 10 라우팅 프로토콜과 커널

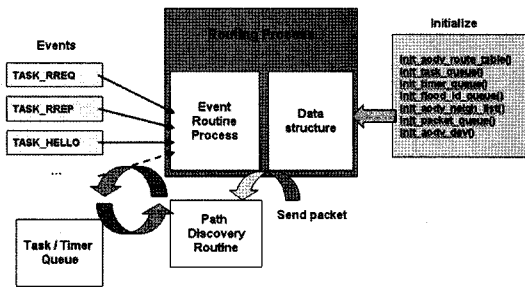


그림 11 라우팅 프로토콜의 전체 프로세스

초기화 루틴은 그림 11에 나타난 초기화 함수들을 통해 라우팅 프로세스에 필요한 데이터 구조와 각 데이터를 관리하기 위한 리스트, 그리고 클러스터 헤더와 멤버 노드를 비롯한 장치에 대한 데이터를 초기화한다.

이벤트 처리 루틴은 태스크 큐에 작업을 할당하고, 프로세스가 wakeup 할 때마다 큐에 있는 작업을 수행하는 방식으로 동작한다. 각 태스크는 라우팅 컨트롤 메시지의 처리 작업과 라우팅 정보를 유지하기 위한 각종 테이블 관리 작업들을 정의한다.

경로 탐색 루틴은 메시지를 전송 할 때 라우팅 테이블에 해당 경로가 존재하지 않을 경우 수행된다. 우선, 전송하려는 메시지를 저장한 후, RREQ(Route REQuest) 메시지를 생성하고 전송한다. 이후의 재전송 등의 작업을 위해 태스크 큐에 RREQ 재전송 태스크를 저장한다. RREQ 메시지의 응답인 RREP 메시지 처리는 이벤트 처리 루틴을 따른다.

또한, 클러스터 기반의 라우팅을 위해서는 클러스터 간의 인접한 노드의 주소가 요구된다. 이를 위해서 주기적인 HELLO 메시지를 이용한다. Ad-hoc 네트워크에서는 특성상 노드의 이동이 잦기 때문에, 주기적인 HELLO 메시지를 이용하여 자신의 위치를 주변 노드에 알려지게 된다. 이 때, 자신의 주소 외에 인접한 클러스터 노드의 주소를 전달하여 클러스터 헤더가 다른 클러스터와 연결 될 수 있게 한다. 인접 노드의 주소를 전

달하는 오버헤드를 고려하여 전송되는 주소는 클러스터 주소를 가리키는 prefix만을 포함한다.

프로토콜에서 사용하는 메시지에는 목적지까지의 경로를 요청하기 위한 RREQ, RREQ에 대한 응답으로 경로를 알려주기 위한 RREP, 경로 상의 에러를 알려주기 위한 RERR, 그리고 앞에서 언급한 자신의 위치와 인접 노드의 주소를 알려주기 위한 HELLO 메시지가 있다. 이들 메시지를 수신하면 각각의 메시지 타입에 따라 앞서 말한 9 가지의 태스크 중 해당되는 태스크가 태스크 큐에 저장되고, 이벤트 처리 루틴에 따라 이를 수행한다.

라우팅 프로세스는 해당 노드의 역할에 따라 다르게 동작한다. 클러스터 헤더는 실질적인 라우팅 프로세스의 주체가 되어 대부분의 라우팅 프로세스를 수행하지만, 클러스터 멤버는 인접 노드의 정보를 전달하는 역할만을 수행하게 된다.

라우팅 프로토콜은 크게 세 부분으로 나뉘어 동작한다. 실제 구현에서는 3 개의 쓰레드를 이용하여 각 부분에 대해 작업을 수행한다.

먼저, 응용 프로그램으로부터 온 데이터를 전송할 때, 목적지(destination)에 해당하는 엔트리가 라우팅 테이블에 존재하지 않는 경우, 즉 목적지까지의 경로가 확보되지 않는 경우에 응용 프로그램으로부터 전송 받은 데이터는 경로가 확보될 때까지 큐에 저장되고 경로 생성을 위한 경로 탐색 작업을 수행한다.

두 번째로 다른 노드로부터 RREQ나 RREP, RERR과 같은 라우팅 컨트롤 메시지를 받은 경우에 해당 메시지에 따른 처리를 수행한다. 이 경우 메시지 타입에 따라 태스크를 생성하여 이벤트 큐에 추가하고, Sleep 상태의 메인 쓰레드를 깨워서 큐에 저장된 작업을 바로 처리하게 한다.

마지막으로 주기적인 HELLO 메시지의 전송으로 자신의 위치를 주변 노드에게 알리고, 자신의 주소와 자신의 인접 노드들의 주소를 클러스터 헤더에 알린다. 또한, 라우팅 테이블과 이웃 노드 테이블을 비롯한 각종 테이블과 리스트의 관리를 수행한다.

최종 완성된 프로그램의 수행 화면은 그림 12와 같다.

6. 결론

RFID 기반의 물류창고 시스템은 RFID 기술을 이용하여 복잡한 물류창고의 작업을 간편하고 빠르게 만들어준다. 이를 더 효율적으로 지원하기 위해 본 논문에서는 클러스터링, IPv6 기반의 주소 지정 및 이동성 지원, 그리고 클러스터 기반의 계층적 라우팅 프로토콜의 3가지 부분으로 나누어 무선 센서 네트워크를 구축하였다.

본 논문은 클러스터링, 주소지정 및 이동성, 그리고 라우팅의 3부분을 각각 구현함으로써 센서 네트워크를

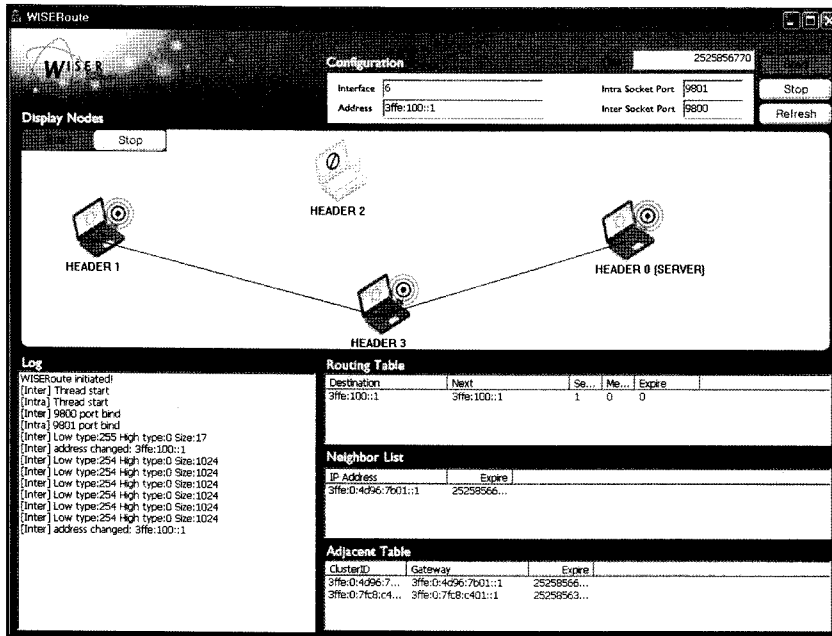


그림 12 프로그램 실행 화면

실제로 구축하였다는데 그 의의를 둔다.

현재 802.11을 사용하여 구축되어 있는 네트워크를 좀더 다양한 MAC을 사용할 수 있도록 실험하고 수정할 예정이다.

참고 문헌

- [1] Weinstein. R., "RFID: a technical overview and its application to the enterprise," IT Professional Volume 7, Issue 3, May-June 2005 Page(s): 27-33, 2005.
- [2] Yang Dayu and Zou Peng, "Event Driven RFID Reader for Warehouse Management," sixth International conference on PDCAT 2005, Pages(s): 895-899, 2005.
- [3] S. Deering and R. Hinden, "Internet Protocol, version 6 (IPv6) specification," IETF RFC2460, December 1998.
- [4] IEEE Std. 802.11, IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications (With Amendment 1-4), 2003.
- [5] S. Basagni, "Distributed Clustering for Ad Hoc Networks," International symposium on Parallel Architectures, Algorithms and Networks, Page(s): 310-315, June 1999.
- [6] D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," IEEE Transaction on Communications, Vol.29, No.11, Page(s): 1694-1701, November 1981.
- [7] M. Charterjee, s. K. Das, and D. Turgut, "WCA: A Weighted clustering Algorithm for Mobile Ad hoc Networks," Journal of Cluster Computing, Special issue on Mobile Ad hoc Networking, No. 5, Page(s): 193-204, 2002.
- [8] C. E. Perkins, "Mobility Support in IPv6," in Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MOBICOM'96), November 1996.
- [9] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6," Draft-ietf-mobileip-ipv6-24.txt, IETF Mobile IP Working Group, December 2003.
- [10] 이신형, 우수창, 유혁, "RFID 물류창고 시스템을 위한 재구성 가능한 클러스터 기반의 라우팅 기법," 한국정보처리학회 추계학술대회, 2005년 11월.
- [11] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum et L. Viennot, "Optimized Link State Routing Protocols for Ad Hoc Networks," IEEE INMIC Pakistan, 2001.
- [12] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers," Proc. of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications, Aug 1994, pp. 234-244.
- [13] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999.
- [14] C. E. Perkins, E. M. Royer, and S. Das, "Ad hoc On-demand Distance Vector (AODV) Routing,"

RFC 3561, July 2003, Category: Experimental, work in progress. <http://www.ietf.org/rfc/rfc3561.txt>

- [15] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing (Kluwer Academic, 1996) chapter 5, pp.153-181.
- [16] M. Jiang, J. Li and Y. C. Tay, "Cluster based routing protocol (CBRP)," IETF MANET Working Group, Internet-Draft, July 1999.
- [17] 우수창, 이신형, 유혁, "RFID 물류창고 시스템을 위한 애드 혹 라우팅 프로토콜에 관한 연구", 한국정보처리학회 추계학술대회, 2005년 11월.
- [18] 이치영, 이신형, 유혁, "RFID 물류 시스템을 위한 계층적 클러스터 기반의 라우팅 프로토콜의 설계 및 구현".



이 신 형

2001년 고려대학교 지구환경과학과 학사
2005년 고려대학교 컴퓨터학과 석사. 2007년 고려대학교 컴퓨터학과 박사 수료. 관심분야는 센서네트워크, RFID, 네트워크 프로토콜, 운영체제



이 치 영

2006년 2월 충남대학교 전기정보통신공학부 컴퓨터전공 학사. 2008년 2월 고려대학교 컴퓨터학과 석사. 관심분야는 IP 네트워크, 다중 경로 라우팅, 운영체제



김 동 신

2003년 2월 고려대학교 컴퓨터학과 졸업
2003년 3월~현재 고려대학교 컴퓨터학과 석박사 통합과정. 관심분야는 QoS provisioning, RFID system, Pervasive Computing in Wireless Sensor Networks



이 찬 행

2004년 한신대학교 정보시스템공학과 학사. 2006년 한신대학교 대학원 컴퓨터정보학과 석사. 2006년~현재 고려대학교 컴퓨터학과 박사과정. 관심분야는 802.11 LAN, 무선 네트워크, 모바일 애드혹 네트워크, 매체 접근 제어 기술, 라우팅 프로토콜, 에너지 효율성



이 원 준

1989년 서울대학교 컴퓨터공학과 학사
1991년 서울대학교 컴퓨터공학과 석사
1998년 Stanford Research Institute (SRI), Research Associate. 1999년 University of Minnesota 컴퓨터공학 박사. 2002년~현재 고려대학교 정보통신대학 컴퓨터학과 부교수. 관심분야는 네트워크, 무선이동통신



민 성 기

1988년 고려대학교 컴퓨터학과 학사. 1989년 런던대학교 컴퓨터학과 석사. 1994년 런던대학교 컴퓨터학과 박사. 1994년~2000년 LG 정보통신 중앙연구소 책임연구원. 2001년~현재 고려대학교 컴퓨터학과 부교수. 관심분야는 분산시스템, 모바일 네트워킹, 라우팅 프로토콜, 모바일 핸드오버, 광대역 통합망, Enhanced QoS



유 혁

1982년 고려대학교 전자공학과 학사. 1984년 서울대학교 전자공학과 석사. 1986년 University of Michigan 전산학 석사. 1990년 University of Michigan 전산학 박사. 1990년~1995년 Sun Microsystems Lab. 1995년~현재 고려대학교 정보통신대학 컴퓨터학과 교수. 관심분야는 운영체제, 멀티미디어, 네트워크