

# 무선 멀티미디어 센서 네트워크에서의 신뢰성 있는 비동기적 이미지 전송 프로토콜

이 좌 형<sup>†</sup> · 선 주 호<sup>††</sup> · 정 인 범<sup>†††</sup>

## 요 약

최근 하드웨어의 발달로 무선 센서네트워크를 이용하여 멀티미디어 데이터를 수집하기 위한 멀티미디어 센서네트워크에 관한 연구가 활발히 이루어지고 있다. 멀티미디어 데이터들은 텍스트 형태의 데이터들에 비해 크기가 매우 크며 데이터를 구성하는 일부의 패킷이 손실되면 전체 데이터가 쓸모가 없어지는 경우가 많아 데이터간 관련성이 매우 크다. 따라서 멀티미디어 데이터를 전송하기 위해서는 신뢰성이 보장되어야만 한다. 하지만 기존의 텍스트 기반 데이터를 위한 전송프로토콜들은 대부분 신뢰성보다는 네트워크의 효율성을 목적으로 연구되어져 멀티미디어 데이터를 전송하기에 적합하지 않다. 본 논문에서는 센서네트워크에서의 신뢰성 있는 비동기적 이미지 전송 프로토콜인 RAIT를 제안한다. RAIT는 네트워크 혼잡으로 인한 패킷 손실을 방지하기 위하여 노드간 이미지 전송시 이중 슬라이딩 윈도우 기법을 적용한다. 노드간 통신 장애로 인한 패킷 손실을 방지하기 위한 수신큐를 위한 슬라이딩 윈도우뿐만 아니라 혼잡으로 인한 패킷손실이 발생하는 전송큐를 위한 슬라이딩 윈도우를 통하여 이미지 전송의 신뢰성을 보장한다. 상위노드는 하위노드들을 이미지별로 비선점형으로 스케줄링하여 패킷손실을 없애면서 노드간 형평성을 높인다. 이중슬라이딩 윈도우를 구현하기 위하여 RAIT에서 라우팅 레이어와 큐 레이어를 제어하도록 하는 크로스레이어 기법을 적용한다. 실험을 통하여 RAIT가 기존 프로토콜에 비해 신뢰성 있게 이미지 전송을 보장함을 보인다.

키워드: 멀티미디어센서네트워크, 신뢰성, 이미지, 이중 슬라이딩 윈도우, 크로스 레이어

## Reliable Asynchronous Image Transfer Protocol In Wireless Multimedia Sensor Network

Joa-Hyoung Lee<sup>†</sup> · Ju-Ho Seon<sup>††</sup> · In-Bum Jung<sup>†††</sup>

## ABSTRACT

Recently, the advance of multimedia hardware has fostered the development of wireless multimedia sensor network which is able to ubiquitously obtain multimedia content such as image or audio from the environment. The multimedia data which has several characteristics such as large size and correlation between the data requires reliability in transmission. However, the existing solution which take the focus on the efficiency of network mainly, is not appropriate to transmit the multimedia data. In the paper, we proposes a reliable asynchronous image transfer protocol, RAIT. RAIT applies double sliding window method in node-to-node image transfer to prevent the packet loss caused by network congestion. The double sliding window consists of one sliding window for the receiving queue, which is used for prevention of packet loss caused by communication failure between nodes and the other sliding window for the sending queue which prevents the packet loss caused by network congestion. the routing node prevents the packet loss and guarantees the fairness between the nodes by scheduling the packets based on the image non-preemptively. The RAIT implements the double sliding window method by cross layer design between RAIT layer, routing layer, and queue layer. The experiment shows that RAIT guarantees the reliability of image transmission compared with the existing protocol.

Key Words: WMSN, Image, Reliability, Double sliding window, Cross-layer

## 1. 서 론

최근 저가의 CMOS 이미징 센서가 널리 보급되고 센서 노드의 하드웨어 성능이 향상되면서 무선 센서네트워크를 이용하여 멀티미디어 수집하는 멀티미디어 센서 네트워크(WMSN, Wireless Multimedia Sensor Network) 분야가 주

\* 본 연구는 산업자원부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.  
† 준 회원: 강원대학교 컴퓨터정보통신공학과 박사과정  
†† 준 회원: 강원대학교 컴퓨터정보통신공학과 학사과정  
††† 통신회원: 강원대학교 컴퓨터정보통신공학전공 교수(교신저자)  
논문접수: 2008년 4월 4일  
수정일: 2008년 5월 21일  
심사완료: 2008년 5월 23일

목받고 있다. 멀티미디어 센서네트워크가 매우 많은 정보를 제공할 수 있는 장점이 있지만 많은 정보를 제공하기 위해서는 많은 연구가 필요하다. 매우 많은 제한을 가지는 센서 노드에서 매우 많은 자원을 필요로 하는 멀티미디어 데이터를 처리하기 위해서는 많은 분야에서 새로운 연구가 이루어져야만 한다[1-3].

에너지가 제한적인 센서노드에서 대용량의 이미지 데이터를 전송하기 위해 많은 패킷을 전송하면 에너지의 빠른 고갈로 네트워크의 수명이 단축될 수 있으며 많은 패킷들이 집중되면서 네트워크에 혼잡을 야기시킬 수도 있다. 이러한 문제점들을 해결하기 위해 에너지 효율적인 전송프로토콜이나 혼잡을 제어하는 프로토콜들이 적용될 수 있다. 하지만 기존의 프로토콜들은 이미지 데이터의 특성을 고려하지 않고 설계되었기 때문에 이미지 전송을 위한 특화된 전송 프로토콜이 필요하다[4,5,6,7].

이미지 데이터의 여러 특성들 중에서 반드시 고려하여야 할 사항 하나가 데이터의 손실이 발생하면 데이터의 품질이 급격히 떨어진다는 것이다. 이미지 데이터는 2차원으로 이루어져 비트스트림이 매우 길며 비트스트림의 일부가 손실되면 이미지가 심하게 손상되거나 사용할 수 없는 상태가 되기도 한다. 하나의 이미지가 사용자에게 제공되려면 이미지를 이루는 모든 비트스트림이 싱크로 전송되어야만 한다. 따라서 센서노드로부터 싱크노드로 이미지를 전송할 신뢰성 있는 이미지 전송프로토콜이 필요하다[8,9,10,11].

신뢰성 보장을 위한 기존의 프로토콜들은 주로 노드간 통신에서의 신뢰성 보장에 치중되어 왔다. 센서노드들이 동작하는 무선환경이 에러가 발생할 가능성이 매우 높기 때문에 전송중에 발생하는 통신에러로 인한 데이터 손실이 많이 발생한다. 기존 프로토콜들은 인터넷에서 사용되던 재전송 기법이나 에러복구 기법들을 도입하여 이런 손실을 회복하려는데 중점을 두고 있다. 이러한 프로토콜들이 노드간 통신에서 신뢰성을 보장할수는 있지만 노드내에서 발생하는 데이터 손실을 회복하기는 힘들다[4,5,6].

본 논문에서는 무선 센서네트워크에서 신뢰성 있는 이미지를 전송하기 위한 프로토콜인 RAIT(Reliable Asynchronous Image Transfer protocol)를 제안한다. RAIT는 하나의 이미지를 구성하는 모든 패킷들을 완전하게 전송하는 것을 목표로 한다. 이를 위하여 이중 슬라이딩 윈도우를 이용하여 노드간 패킷 전송에서 혼잡으로 인한 패킷 손실이 발생하지 않도록 한다. 이중 슬라이딩 윈도우는 수신노드의 수신큐를 위한 슬라이딩 윈도우와 수신노드의 전송큐를 위한 슬라이딩 윈도우로 구성된다. 하위노드는 이미지를 전송하기 위해서는 상위노드에 전송허가를 받아야 하며 전송허가를 받은 노드는 상위노드에 독점적으로 패킷을 전송할 수 있다. RAIT는 크로스 레이어(Cross-Layer)를 통하여 큐와 라우팅을 조절한다. RAIT에서는 상위노드로부터 허가를 받은 경우에만 전송이 가능하도록 라우팅 레이어를 제어하고 가용한 큐의 크기를 얻어오고 큐의 전송을 조절하기 위해 큐 레이어를 제어한다[12-15].

본 논문의 구성은 다음과 같다. 2장에서는 흐름제어 기법

중 하나인 Token-Bucket 방식에 대해 알아본다. 3장에서는 RAIT를 이루고 있는 각 Layer의 구성에 대해, 4장에서는 RAIT의 동작방식을 알아본다. 5장에서는 시뮬레이터를 이용하여 RAIT의 성능을 평가하며, 6장의 결론 및 향후연구로써 글을 맺는다.

## 2. 관련연구

무선 네트워크에서 데이터 전송의 신뢰성을 보장하기 위한 많은 연구가 진행되었지만, 이를 그대로 무선 센서네트워크에 적용시키는 것은 적합하지 않다[3]. 무선통신에서 데이터 신뢰성을 유지하기 위한 기법으로 알려져 있는 PSFQ[4]와 RMST[5]는 NAK(Negative Acknowledgement)와 ARQ(Automatic Repeat Request)를 사용하여 노드간 통신에서 발생하는 에러를 복구한다. FEC(Forward Error Correction)는 에러정정코드를 사용하여 데이터 에러를 복구한다. 이미지 전송을 위한 FEC 기반의 UEP(Unequal Error Protection)는 무선통신에서의 에러를 줄이고 효과적인 전송을 가능하게 한다[6, 7, 8]. 패킷의 손실이 감지되면 FEC 알고리즘은 중요 데이터를 복구하기 위해 FEC code를 적용한다. 이러한 기법들은 노드간에 발생하는 패킷에러는 복구할 수 있지만 앞서 언급하였듯이 혼잡으로 인해 발생하는 노드내에서의 패킷 손실은 복구할 수 없다.

ESRT[19]나 CODA[18]와 같은 혼잡제어 프로토콜들은 혼잡으로 인하여 노드 내에서 발생하는 패킷 손실을 제어하기 위해 혼잡 발생시 하위 노드들에서 데이터를 보고하는 주기를 늘려 패킷량을 줄임으로써 패킷손실을 방지하는 방법을 사용한다. 온도나 습도와 같은 스칼라 형태의 1차원 데이터들은 샘플링 주기를 조절하더라도 데이터 분석에 크게 영향을 미치지 않기 때문에 이러한 혼잡제어 프로토콜들이 사용될 수 있다. 하지만 이미지의 경우 한번의 샘플링을 통해 얻어지는 데이터가 매우 많기 때문에 이런 프로토콜들을 적용하기 어렵다.

이미지 데이터의 높은 신뢰성을 유지하기 위한 다른 한 가지 방법은 Multi-path 방식의 전송이다[9]. Multi-path 방식은 두 개의 센서노드에 동일한 데이터를 전송함으로써 통신 에러나 노드 장애로 인해 발생하는 손실로 인한 재전송을 최소화한다. 이러한 분할 스키마는 링크 채널 저하 및 노드에러 발생에서 일정 수준 이상의 이미지 품질을 유지해준다. 하지만 매우 제한적인 에너지를 갖는 센서네트워크에서 같은 데이터를 중복 전송한다는 것은 에너지의 낭비를 의미한다. 신뢰성을 보장하는 것도 중요하지만 에너지 소모가 심해진다면 네트워크 수명을 단축시키는 결과를 초래할 수 있다.

## 3. RAIT

무선 센서 네트워크 상에서 패킷이 손실되는 것은 크게 두가지 경우로 나눌 수 있다. 첫 번째로 무선 통신환경이

열악하기 때문에 노드간 전송시 패킷에 오류가 발생하여 손실되는 경우이다. 이런 문제를 해결하기 위해 오류가 발생한 패킷을 복구하는 기법이나 재전송하는 기법들이 사용된다. 두 번째로는 네트워크에 혼잡이 발생하여 노드상의 큐에 오버플로우가 발생하여 패킷이 손실되는 경우이다. 이 문제를 해결하기 위해서는 노드별로 전송하는 패킷을 조절하는 기법이 필요하다. RAIT는 두 가지의 패킷 손실을 방지하기 위해 이중 슬라이딩 윈도우 기법을 사용한다.

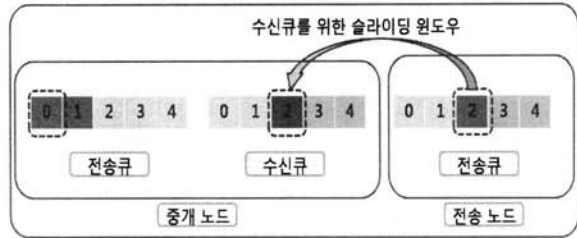
본 논문에서는 센서노드들이 싱크노드를 중심으로 트리형태의 라우팅 경로를 설정하는 라우팅 프로토콜을 가정한다. 일반 센서노드들은 주기적으로 이미지를 센싱하여 싱크노드로 전달하며 싱크노드는 주기적으로 라우팅 패킷을 전송하여 라우팅 경로를 유지한다. 센서노드들은 홉카운트와 링크상태 등에 따라 경로를 설정한다. 트리형태의 라우팅에서는 여러 개의 자식노드들이 하나의 부모노드를 공유하여 사용한다. 자식노드들은 패킷을 전송하기 위해 경쟁적으로 상위노드에 접근한다.

### 3.1 수신큐를 위한 슬라이딩 윈도우

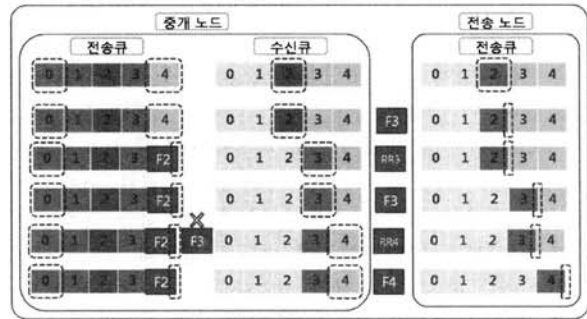
한 번에 전송할 수 있는 패킷수를 제한함으로써 큐 오버플로우를 줄이는 슬라이딩 윈도우 개념이 혼잡제어에 많이 사용된다. 슬라이딩 윈도우에서는 한 번에 전송할 수 있는 패킷수를 윈도우로 정의하고 윈도우 내에 있는 패킷들은 제한없이 보낼 수 있다. 상대방으로부터 수신확인 패킷을 받으면 윈도우를 조절하여 전송량을 조절한다. 일반적으로 양 끝단 간에 패킷을 주고받을 때 슬라이딩 윈도우를 적용하기 때문에 고려하여야할 큐는 패킷을 전송하는 노드의 전송큐와 패킷을 수신하는 노드의 수신큐이다.

센서네트워크에서도 노드간 통신에 큐가 사용된다. 자원이 제한적인 센서노드에서는 수신큐 보다는 전송큐에 많은 양의 메모리를 할당한다. 보통 수신큐는 크기가 1, 즉 패킷 하나를 수신할 수 있는 큐가 사용되며 전송큐는 수십에서 수백개의 패킷을 저장할 수 있는 큐가 사용된다. 노드들은 패킷 전송시 무선 환경에서 다른 노드와 경쟁하여야 하기 때문에 채널 접근 시간이 길어질 수 있으며 오류가 발생한 패킷을 재전송하기 위해 패킷을 임시 저장하여야 하기 때문에 하나의 패킷을 완전히 전송하는데 오랜 시간이 걸릴 수 있어 수신큐보다 전송큐를 크게 설정한다. 따라서 센서네트워크에서의 슬라이딩 윈도우에서는 크기가 1인 윈도우가 사용된다. (그림 1)은 중개노드와 전송 노드 간에 크기가 1인 슬라이딩 윈도우를 보여준다.

노드간 1대1 통신인 경우에는 전송큐와 수신큐 사이의 슬라이딩 윈도우로 패킷 손실을 방지할 수 있다. 하지만 패킷이 멀티홉을 거쳐 전송되는 경우에는 패킷을 중개하는 노드가 라우터처럼 동작하기 때문에 고려하여야할 큐가 늘어나며 따라서 추가적인 슬라이딩 윈도우가 필요하다. 패킷을 중개하는 라우팅 노드는 수신한 패킷을 자신이 소모하는 것이 아니라 상위 노드로 전달해야 하기 때문에 자신의 전송큐로 패킷을 삽입한다. 따라서 고려하여야할 큐는 전송노드의 전송큐, 중개 노드의 수신큐, 그리고 중개노드의 전송큐



(그림 1) 단일 슬라이딩 윈도우



(그림 2) 큐 오버플로우

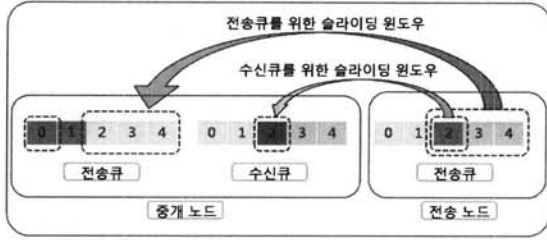
이렇게 3개이다.

대부분의 프로토콜들에서 중개노드의 전송큐는 프로토콜 설계의 고려대상이 아니다. 하지만 중개노드의 전송큐를 고려하지 않기 때문에 중개노드에서 큐 오버플로우가 발생하며 이로 인하여 패킷이 손실된다.

(그림 2)는 단일 슬라이딩 윈도우만을 사용하는 경우에 패킷이 손실되는 예를 보여준다. 수신큐의 크기가 1이기 때문에 슬라이딩 윈도우(점선)이 하나의 패킷만을 포함하고 있다. 전송노드가 패킷 F2를 전송하면 중개노드는 수신한 패킷을 전송큐로 전달하고 전송노드로 다음 패킷을 전송하라는 RR3 패킷을 전송한다. RR3패킷을 수신한 전송노드는 다음 패킷인 F3패킷을 전송한다. F3패킷을 수신한 중개노드는 F3패킷을 전송큐로 전달하고 RR4패킷을 전송한다. 하지만 이 상황에서 중개노드가 패킷을 상위노드로 전송하지 못하여 그림에서 처럼 전송큐가 가득찬 경우 F3패킷은 전송큐에 삽입되지 못하고 손실된다. 중개노드의 수신큐와 전송큐 사이에서 손실되는 패킷은 단일 슬라이딩 윈도우로 복구하지 못한다. 수신큐에서 이미 RR패킷을 전송한 후이기 때문이다.

### 3.2 이중 슬라이딩 윈도우

이러한 노드상의 수신큐와 전송큐 사이에 발생하는 패킷 손실을 방지하기 위하여 본 논문에서는 이중 슬라이딩 윈도우를 사용한다. 첫 번째 슬라이딩 윈도우는 앞서 설명한 전송노드의 전송큐와 중개노드의 수신큐사이에 적용된다. 두 번째 슬라이딩 윈도우는 전송노드의 전송큐와 중개노드의 전송큐 사이에 적용된다. (그림 3)은 이중 슬라이딩 윈도우를 보여준다. 앞서 설명하였듯이 첫 번째 슬라이딩 윈도우에는 크기가 1인 윈도우가 사용되지만 두 번째 슬라이딩 윈도우의 윈도우 크기는 전송큐의 크기에 따라 동적으로 변한다.

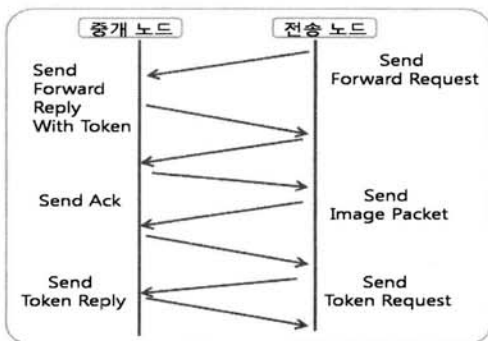


(그림 3) 이중 슬라이딩 윈도우

센서노드에서 패킷을 수신하면 CRC 기법과 같은 오류확인 기법으로 패킷에 오류가 발생하였는지 확인하며 오류가 없는 경우 이에 대한 확인패킷(Ack패킷)을 전송하여 수신을 통보한다. 이 작업은 주로 소프트웨어로 구현되지만 최근에는 하드웨어 기술의 발전으로 하드웨어 상에서 빠르게 처리되기도 한다. 이 확인패킷을 통하여 수신큐를 위한 슬라이딩 윈도우의 크기가 조절된다. 수신큐를 위한 슬라이딩 윈도우는 크기가 1이기 때문에 별도로 윈도우의 크기를 협의할 필요가 없으며 가용한 큐의 크기를 전송할 필요도 없다. 하지만 전송큐를 위한 슬라이딩 윈도우는 크기가 가변적이기 때문에 전송노드와 중개노드 간에 큐에 대한 정보를 주고 받을 필요가 있다.

### 3.3 토큰 협약

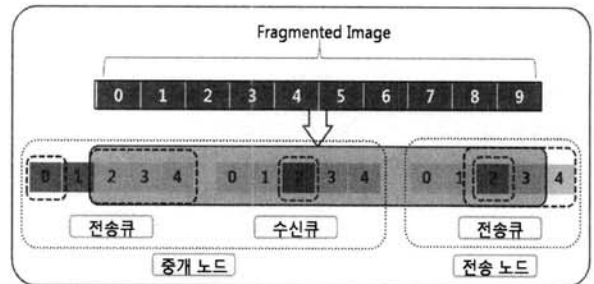
슬라이딩 윈도우에서는 두 노드 간에 한번에 전송가능한 토큰의 양을 협약할 필요가 있다. 상위노드가 수신한 패킷이 전송큐에 삽입될 때마다 하위노드로 확인패킷을 전송하면 오버헤드가 매우 심하게 발생하기 때문에 RAIT에서는 Two-Way Handshake 기법을 사용하여 전송허가와 전송가능한 패킷 수를 주고 받도록 한다. TCP와 같은 프로토콜에서는 양방향 통신을 하기 때문에 Three-Way Handshake 기법을 사용하지만 센서네트워크에서는 단방향 통신을 하기 때문에 Two-Way Handshake로 충분하다. 전송 가능한 패킷 수를 토큰(Token)으로 정의하여 중개노드가 전송노드에 토큰을 할당하도록 한다. 전송노드가 할당받은 토큰을 모두 사용하면 중개노드로 토큰 요청 패킷을 전송하여 토큰을 할당받도록 한다. (그림 4)는 전송노드와 중개노드 사이에 토큰을 할당받는 것을 보여준다.



(그림 4) 토큰할당

### 3.4 비선점형 스케줄링

여러 노드가 균등하게 패킷을 전송하기 위해서는 하나의 노드가 한번에 전송 가능한 패킷의 양에 제한을 두어야 한다. 부모노드는 패킷의 양으로 자식노드들을 스케줄링하여야 한다. RAIT에서는 하나의 이미지를 스케줄링의 단위로 한다. 하나의 이미지를 구성하는 패킷들이 서로 다른 시점에 전송되면 하나의 이미지를 구성하는 패킷들 간에 지연이 발생할 수 있기 때문에 패킷을 취합하는 싱크노드에서 버퍼링을 위한 메모리를 추가적으로 할당해야하는 등에 문제가 발생하기 때문에 하나의 이미지를 구성하는 패킷들은 연속적으로 처리될 필요가 있다. RAIT에서는 노드간 통신에서 토큰 요청에 제한을 두어 하나의 이미지를 구성하는 패킷들을 모두 연속적으로 전송할 수 있도록 한다. 상위노드로 이미지 패킷을 전달하기 위해서는 상위노드로부터 전송허가를 받아야 하며 이 허가는 하나의 이미지를 구성하는 패킷들을 모두 전송할 때까지 유효하다. 하나의 하위노드에 패킷 전송을 허가한 상위노드는 다른 하위노드들에게서 오는 허가요청을 무시한다. 상위노드로부터 전송허가를 받지 못한 노드는 일정시간 대기하였다가 다시 전송허가를 요청하여야 한다. (그림 5)는 이미지를 구성하는 패킷들이 노드들의 큐를 독점적으로 사용하는 것을 보여준다. 이미지를 구성하는 패킷의 수가 전송큐의 크기보다 매우 클 경우 이미지는 여러 노드에 걸쳐져 있을 수 있다.



(그림 5) 이미지 전송 보장

### 3.5 크로스 레이어 (Cross-Layer)

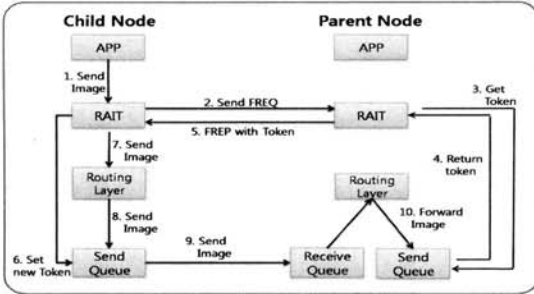
상위노드와 하위노드 간에 한 이미지를 이루는 모든 패킷을 완전하게 전송하려면 RAIT에서 하위 레이어인 라우팅 레이어와 큐 레이어의 동작을 제어할 필요가 있다. 한 이미지의 패킷들은 모두 하나의 상위노드로 전송되어야 하기 때문에 상위노드로부터 전송허가를 받은 이후에는 새로운 부모노드를 선택하지 못하도록 라우팅 레이어를 설정하여야 한다. 상위노드로부터 수신한 토큰에 크기만큼만 전송하도록 큐레이어에 대한 전송제한이 필요하다. (그림 6)은 RAIT에서 크로스 레이어를 통해 이미지가 전송되는 과정을 보여준다.

## 4. 성능평가

### 4.1 실험환경 및 평가요소

본 논문에서 제안하는 RAIT의 성능을 알아보기 위해 몇





(그림 6) 크로스 레이어 디자인

티 홉 라우팅 프로토콜인 MintRoute 와의 이미지 전송률 차이를 알아보았다. 이를 위해 시뮬레이터인 TOSSIM과 TinyViz, 스크립트 언어인 Tython을 이용하였다. Tython은 TOSSIM에서 사용되는 Java와 Python으로 구성된 스크립트 언어이며 TOSSIM과 연동하여 센서노드를 제어하거나, 센서 어플리케이션의 데이터 값을 얻어오는 등 다양한 기능을 제공한다. 노드들을 12 X 12로 배치하여 총 144개의 노드를 배치하였으며 노드당 총 8개의 이웃 노드를 갖도록 설정하였다. 실험에 따라 1 ~ 20개 노드의 임의로 선택된 이미지 노드가 각각 총 10개의 이미지를 전송하도록 하였으며 이미지별 패킷수는 54개로 고정하였다. 센서노드의 수신큐는 1로 설정하였으며 전송큐는 32개로 설정하였다. RAIT는 이미지 전송시 일정시간 동안 상위노드를 선점하여 사용하는 방식이기 때문에 동시에 여러 노드에서 패킷 전송을 하려는 경우 지연이 발생할 수 있다. 이를 확인하기 위하여 노드별 동기화 간격을 0 ~ 40초 사이로 변화하면서 결과를 측정하였다[16, 17].

평가항목으로는 이미지 수신율, 총 패킷 수신량, 패킷별 전송시간, 그리고 이미지 수신 소요시간을 비교하였다. 이미지 수신율은 각 노드가 전송한 10개의 이미지들 중 싱크노드에서 54개의 패킷이 모두 완전하게 수신된 비율을 의미한다. 이미지를 구성하는 54개의 패킷중 싱크에서 하나라도 수신하지 못하면 이미지를 수신하지 못한 것으로 간주하였다. 이는 이미지의 특성상 일부의 데이터가 손실되면 전체 이미지가 손실되기 때문이다. 총 패킷 수신율은 싱크노드에서 수신한 모든 패킷수를 나타낸다. 패킷별 전송시간은 하나의 패킷이 전송되는데 소요되는 시간을 의미하며 이미지 수신 소요시간은 이미지를 구성하는 모든 패킷이 전송되는데 소요된 시간을 의미한다.

## 4.2 전송 노드수 변화

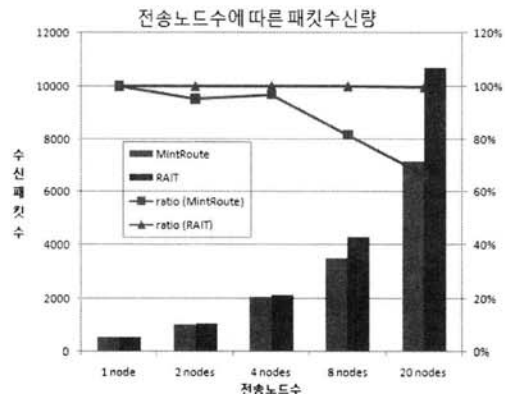
### 4.2.1 수신율

네트워크 상에 이미지를 전송하는 노드에 수가 증가할수록 전송되는 패킷에 양도 많아져 혼잡이 발생할 가능성이 많아지며 결과적으로 중간 노드에서 패킷이 손실될 가능성이 많다. 본 실험에서는 총 144개의 노드중 이미지를 전송하는 노드에 수를 1개, 2개, 4개, 8개 그리고 20개의 노드로 증가시키면서 전체 패킷 수신량과 이미지 수신율을 측정하였다. 실험에서 큐에 크기는 32개로 고정시켰으며 노드간

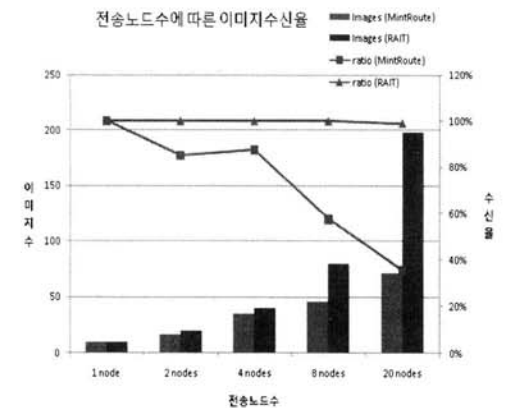
시간 동기화 정도는 40초 범위로 설정하였다.

(그림 7)은 전송노드수를 변화시켰을 때 패킷 수신량과 이미지 수신율을 보여준다. (그림 7)의 (A)에서 보면 전송노드의 수가 적을 경우 수신되는 패킷량이 MintRoute와 RAIT 사이에 큰 차이가 없게 나타나는데 이는 이미지를 전송하는 노드에 수가 적으면 혼잡이 거의 발생하지 않기 때문이다. 전송노드수가 적은 경우에는 MintRoute나 RAIT간에 성능차이가 거의 없는 것으로 나타난다. 이에 반해 전송노드수가 증가하면서 MintRoute에서 수신되는 패킷량은 크게 줄어들어 20개의 노드의 경우 60%가까이 떨어지는 것으로 나타나 혼잡발생으로 인하여 많은 패킷들이 손실되었음을 알 수 있다. 하지만 RAIT에 경우 전송노드에 수가 20개로 증가하더라도 수신량이 100%를 유지하는 것으로 나타나 혼잡으로 인한 패킷 손실이 발생하지 않음을 확인할 수 있다.

(그림 7)의 (B)에서 이미지 수신율을 비교하면 RAIT가 MintRoute에 비해 월등히 나은 성능을 보임을 알 수 있다. 그림에서 보듯이 MintRoute의 경우에는 전송노드의 수가 1개인 경우에는 모든 이미지를 수신하지만 전송노드에 수가 2개에서 4개로 늘어나면 15%가까이 수신율이 떨어져 80% 정도의 수신율을 보이고 있다. (그림 7)의 (A)에서 MintRoute는 전송노드의 수가 2개와 4개인 경우 패킷 수신



(A) 전송노드수에 따른 패킷 수신량



(B) 전송노드수에 따른 이미지 수신율

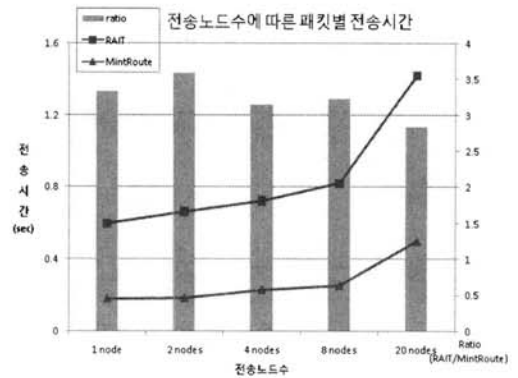
(그림 7) 전송노드수 변화에 따른 패킷 수신량과 이미지 수신율 변화

율은 95%정도로 높았으나 (그림 7)의 (B)에서 이미지 수신율이 85%로 패킷 수신율에 비해 10%정도 낮게 나타나는데 이는 이미지의 특성상 하나의 이미지를 구성하는 전체 패킷 중 하나의 패킷이 손실되더라도 해당 이미지가 손실된 것으로 간주하였기 때문이다. 압축방식에 따라 일부의 패킷이 손실되더라도 이미지 복원이 가능한 경우가 있지만 전체 이미지를 수신한 것에 비해서는 화질과 같은 이미지 품질이 크게 떨어지기 때문에 본 논문에서는 전체 패킷이 수신된 경우에만 이미지를 복원할 수 있는 것으로 간주하였다. 따라서 이미지를 구성하는 전체 패킷들에 대한 전송을 보장하지 않는 MintRoute의 경우 하나의 이미지를 이루는 전체 패킷중 소수의 패킷이 손실되어 전체 이미지가 손실된 것으로 처리되었다. MintRoute의 경우 전송노드수가 8개와 20개로 증가하면서 이미지 수신율이 60%와 35%정도로 심하게 감소하였는데 이는 전송노드수가 증가하면서 혼잡이 증가하고 이로 인하여 이미지를 구성하는 일부의 패킷들의 손실이 증가하였기 때문이다. 전송노드가 20개인 경우 전체 이미지의 3분의 1정도만 전송되면 이미지가 거의 전송되지 않는 것으로 볼 수 있어 MintRoute가 이미지 전송에 적합하지 않음을 알 수 있다. 이에 반해 RAIT는 전송노드수가 증가하더라도 패킷을 100% 모두 수신하기 때문에 이미지 수신율도 전송노드수에 상관없이 100%를 보이고 있다. RAIT는 패킷 전송시 상위노드의 전송큐를 고려하여 패킷을 전송하기 때문에 노드의 큐 오버플로우로 인한 패킷 손실이 발생하지 않도록 하는 것으로 목적으로 하며 실험 결과에서 전송노드의 수가 증가하여 혼잡이 발생하더라도 패킷 손실이 발생하지 않으며 따라서 모든 이미지를 완전하게 전송함을 확인 할 수 있다.

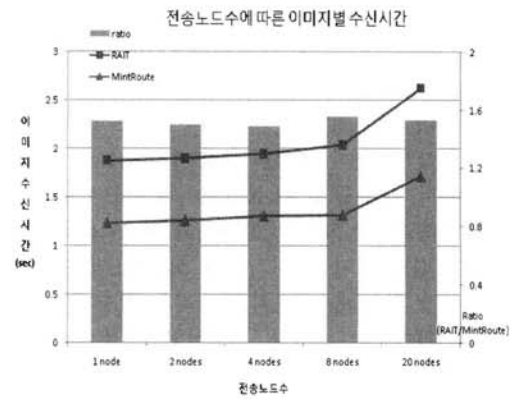
4.2.2 수신시간

RAIT는 이미지를 신뢰적으로 전송하기 위해 중간 노드에서 발생하는 큐 오버플로우를 방지하기 위한 방안으로 하나의 하위노드가 상위노드의 큐를 독점하여 사용하도록 하는데 이로인하여 패킷 전송시 지연이 발생할 수 있다. 이미지 전송노드에서 싱크까지 각각의 패킷이 전송되는데 소요되는 시간과 하나의 이미지가 완전하게 수신되는데 소요되는 시간을 측정하여 (그림 8)에 나타내었다.

(그림 8)의 (A)를 보면 패킷 하나를 전송하는데 걸리는 시간은 전송노드수에 비례하여 증가하고 있는 것으로 나타나며 RAIT가 MintRoute에 비해 3배 이상 걸리는 것으로 나타나고 있다. 하지만 (그림 8)의 (B)에서 보듯이 이미지 전체를 전송하는데 걸리는 시간은 RAIT와 MintRoute사이에 1.5배 정도의 차이만 나는 것으로 나타났는데 이는 MintRoute에서는 패킷들을 이미지별로 구별해서 전송하는 것이 아니라 무작위로 전송하기 때문에 패킷별 전송시간은 짧지만 이미지 전체 전송시간은 길게 걸리며 이에 반해서 RAIT에서는 이미지별 패킷을 한번에 모아서 전송하기 때문에 패킷별 전송시간은 짧지만 패킷들이 연속적으로 전송되어 이미지 전체 전송시간은 짧아지는 특징이 있다. 더군다나 MintRoute의 경우 모든 패킷이 전송되는 것이 아니라 일부 패킷이 손실되는 상황에서



(A) 전송노드수에 따른 패킷별 전송시간



(B) 전송노드수에 따른 이미지별 수신시간

(그림 8) 전송노드수에 따른 패킷별 전송시간 및 이미지별 수신시간

전송시간을 비교한 것이기 때문에 실제 전송시간의 차이는 더욱 줄어들 것으로 판단된다.

4.3 시간 동기화 변화

4.3.1 수신율

많은 어플리케이션들이 동일한 시점에 데이터를 센싱하기를 원하며 이를 위하여 시간동기화에 관한 연구들이 많이 진행되고 있다. 센서노드 간에 시간이 동기화되어 있으면 같은 시점에 데이터를 센싱하기 때문에 데이터 간에 동기화는 이루어질 수 있어 좋으나 여러 개의 노드들이 동시에 패킷을 전송하면 혼잡이 발생하여 패킷이 손실될 가능성이 높아지는 문제점이 있다. 특히 이미지의 경우 한번 센싱으로 전송해야할 데이터의 양이 많기 때문에 노드 간에 동기화가 잘되어 있으면 혼잡이 발생할 가능성도 더욱 높아진다.

본 실험에서는 노드간에 시간동기화를 0초, 1초, 10초, 20초 그리고 40초 정도로 설정하여 이미지를 전송하도록 하였다. 시간 동기화 정도는 노드 간에 최대 해당 시간만큼 시간차가 있다는 것을 의미한다. 노드 간에 시간동기화 차이가 클수록 노드들이 동시에 이미지를 전송할 가능성이 적어져 혼잡이 발생할 가능성이 낮아지므로 이미지 수신율이 높아질 수 있다. 동기화 실험에서는 20개의 노드가 이미지를 전송하도록 하였으며 큐 크기는 32개로 하였다. 노드간 시간동기화는 이미지 전송노드 뿐만 아니라 전체 센서노드들

사이에 동기화 정도를 나타낸다.

(그림 9)는 노드간 시간 동기화에 따른 메시지 수신량과 이미지 수신량을 보여준다. MintRoute의 경우 동기화 간격이 좁을수록 메시지 수신량이 40%정도로 매우 적게 나타나고 있다. 이는 메시지를 100% 모두 수신하는 RAIT에 비해 매우 낮은 수치이다. 동기화 간격이 길어지면서 메시지 수신량이 증가하고 있으나 10초 이후로는 거의 변화가 없는 것으로 나타나고 있다. MintRoute는 동기화 간격이 좁은 경우에 이미지 수신량이 매우 낮아져서 10%정도만 수신하는 것으로 나타나 이미지를 거의 수신하지 못함을 알 수 있다. (그림 9)의 (A)와 (B)를 분석해볼 때 MintRoute는 메시지 수신량이 40%에서 60%정도로 절반정도의 수신율을 보이고 있으나 이미지 수신율은 10%에서 30%정도로 매우 낮게 나타나고 있어 패킷이 중간노드에서 손실되는 문제점뿐만 아니라 싱크노드까지 전송된 패킷들에 대부분도 이미지를 구성하지 못하여 버려지고 마는 문제점을 보이고 있다. 이는 전체적으로 네트워크의 효율성이 매우 낮다는 것을 보여주고 있다. 이에 반해 RAIT는 동기화 정도에 상관없이 모든 메시지와 패킷을 100% 전송하는 것으로 나타나 월등히 나은 성능을 보임을 알 수 있다.

#### 4.3.2 수신시간

(그림 10)은 노드별 시간동기화에 따른 패킷별 전송시간과 이미지별 전송시간을 보여준다. 노드 간에 시간 동기화

간격이 짧아지면 동시에 패킷을 전송하려는 노드들이 많아지기 때문에 경쟁과 충돌이 발생하여 혼잡 발생정도가 커질 수 있다. 혼잡을 처리하는 방식에 따라 전송시간에 차이가 발생할 수 있는데 MintRoute의 경우 혼잡이 발생하면 패킷을 버리기 때문에 동기화 정도에 따라 전송시간이 크게 차이가 나지 않지만 RAIT의 경우 하나의 노드가 패킷을 전송중이면 다른 주위의 노드들은 대기하도록 하여 혼잡을 제거하기 때문에 경쟁이 심해질수록 전송시간이 길어진다.

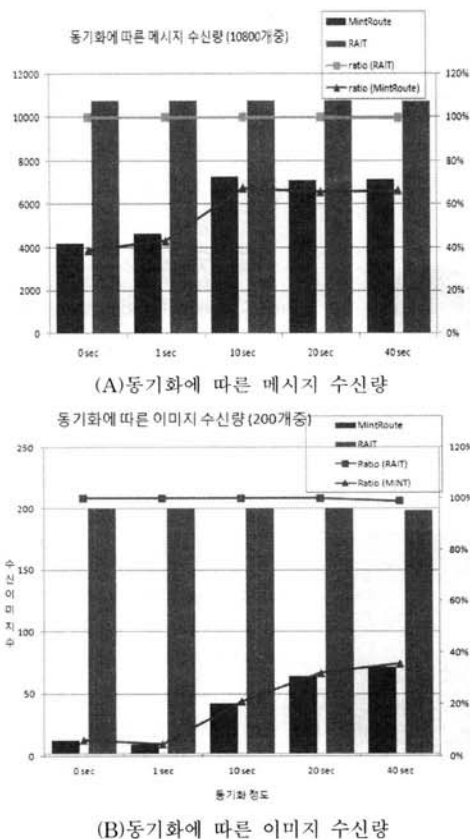
본 실험에서는 RAIT의 경우 하나의 노드가 패킷 전송중이면 다른 노드들은 2초정도 대기하였다가 다시 전송요청 패킷을 보내도록 하였기 때문에 동기화 간격에 따라 2초에서 4초정도 차이가 발생하는 것을 볼 수 있다. 전송 노드수에 비해 대기시간이 매우 길기 때문에 전체적인 전송시간이 길게 나타났으며 대기시간을 이미지 전송시간을 고려하여 설정할 경우 전송시간은 크게 줄어들 것으로 판단된다.

(그림 10)을 보면 RAIT와 MintRoute간에 패킷전송시간에 최대 5배이상 차이가 나는 것으로 나타나 성능 차이가 큰 것으로 생각할 수 있지만 (그림 9)에서 보듯이 MintRoute가 RAIT에 비해 메시지 수신율이 최소 30%정도 밖에 안되기 때문에 동일한 수의 패킷이 전송된다면 패킷 전송시간 차이는 훨씬 줄어들 것으로 판단할 수 있다. 메시지 수신율차가 제일 적은 40초의 경우 패킷별 전송시간과 이미지 전송시간이 모두 비슷하게 나타났으나 수신율은 RAIT가 MintRoute에 비해 두배이상 많은 것으로 나타나 전체적인 전송시간은 RAIT가 MintRoute에 비해 빠른 것으로 생각할 수 있다.

### 5. 결론

최근 무선 센서네트워크에서 멀티미디어 데이터를 사용하기 위한 시도가 이루어지고 있다. 다양한 멀티미디어 데이터중 이미지는 가장 많은 정보를 사용자에게 제공할 수 있는 이점이 있지만 크기가 매우 크다는 문제점을 가진다. 특히 하나의 이미지를 구성하는 데이터들 중 일부가 손실되면 이미지 전체에 영향을 미쳐 사용자가 이미지를 복원하기가 힘들어진다. 따라서 사용자에게 QoS를 보장하기 위해서는 이미지 전송시 신뢰성을 보장하여야 한다.

본 논문에서는 센서네트워크에서 신뢰성있는 이미지 전송을 위한 프로토콜로 RAIT를 제안하였다. RAIT는 노드간 통신장애로 인한 패킷손실과 네트워크 혼잡으로 인한 패킷 손실 모두를 방지하기 위하여 패킷 전송시 이중 슬라이딩 윈도우 기법을 사용하였다. 하위노드는 상위노드로 패킷 전송시 수신큐를 위한 슬라이딩 윈도우를 적용하여 장애 발생시 손실된 패킷을 재전송하도록 하였다. 노드 간에 패킷이 정상적으로 전달되더라도 혼잡이 발생하면 상위노드의 전송큐에서 패킷이 손실될 수 있다. 이를 방지하기 위하여 상위노드의 전송큐를 위한 슬라이딩 윈도우를 적용하여 상위노드의 큐를 독립적으로 사용할 수 있도록 하였다. 상위노드는 하위노드들을 이미지별로 비선점적 스케줄링 기법을 적용하도록 하였다. RAIT를 구현하기 위해서 하위 라우팅 레이어와 큐 레이어를



(그림 9) 동기화에 따른 메시지 수신량과 이미지 수신량

RAIT에서 조절하도록 하는 크로스 레이어 기법을 도입하였다. 실험을 통하여 RAIT가 다른 프로토콜에 비해 완전하게 이미지 전송을 100% 보장함을 보였다.

### 참 고 문 헌

[1] I. F. Akyildiz, T. Melodia, K. R. Chowdhury, "A survey on wireless multimedia sensor networks", *Computer Networks*, 51:921-960, 2007.

[2] S. Mishra, M. Reisslein, G. Xue, "A Survey of Multimedia Streaming in Wireless Sensor Networks", *IEEE Communications Surveys and Tutorials*, 2008.

[3] E. Gurses, O. B. Akan, "Multimedia Communication in Wireless Sensor Networks," *Annals of Telecommunications*, Vol.60, No.7-8, pp.799-827, July-August 2005.

[4] C.Y. Wan, A.T. Campbell, L. Krishnamurthy, "PSFQ: a reliable transport protocol for wireless sensor networks", *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ACM Press, 2002, pp.1-11.

[5] F. Stann, J. Heidemann, "RMST: reliable data transport in sensor networks", *1st IEEE International Workshop on Sensor Net Protocols and Applications(SNPA)*, Anchorage, Alaska, USA, 2003.

[6] P.G. Sherwood, K. Zerger, "Error protection for progressive image transmission over memoryless and fading channels", *IEEE Transactions on Communications* 46(12)(1998) 1555-1559.

[7] A. Mohr, E. Riskin, R. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channel through forward error correction", *IEEE Journal on Selected Areas in Communications* 18(6)(2000) pp.819-828.

[8] K. W. Lee, R. Puri, T. Kim, K. Ramchandran, V. Bhargh-avan, "An integrated source and congestion control frame-work video streaming in the Internet", *Proceedings on Infocom, Tel-Aviv, Israel*, 2000.

[9] H. Wu, A. A. Abouzeid, "Error Robust Image Transport in Wireless Sensor Networks", *Proceeding of 5th workshop on Applications and Services in Wireless Networks(ASWN 2005)*.

[10] H. Wu, A. A. Abouzeid, "Error resilient image transport in wireless sensor networks", *Computer Networks*, 50:2873-2887, 2007.

[11] V. Lecuire, C. Duran-Faundez, N. Krommenacker, "Energy-Efficient Transmission of Wavelet-Based Images in Wireless Sensor Networks," *EURASIP Journal on Image and Video Processing*, Vol.2007, Article ID 47345, 11 pages, 2007.

[12] S. Shakkottai, T. S. Rappaport, C. Karlsson, "Cross-layer Design for Wireless Networks", *IEEE Communications manazine*, October, 2003.

[13] Comer, Douglas E. "Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture", Prentice Hall, 1995. ISBN 0132169878.

[14] Peterson, Larry L. & Davie, Bruce S. "Computer Networks: A Systems Approach", Morgan Kaufmann, 2000.

[15] S. Doley, A. Keizelman, "Non-Preemptive Real-Time Scheduling of Multimedia Tasks", *Real-Time Systems*, Vol.17, Issue 1, 1999.

[16] A. Woo, T. Tong, D. Culler. "Taming the underlying challenges of reliable multihop routing in sensor networks." In *Proceedings of the first international conference on Embedded networked sensor systems*, pp.14 - 27. ACM Press, 2003.

[17] P. Levis, N. Lee, M. Welsh, D. Culler, "TOSSIM:Accurate and Scalable Simulation of Entire TinyOS Applications", *SenSys'03*, November 5-7, 2003.

[18] C. Wan, Shane B. Eisenman, A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks", in *Proc. of First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pp.266-279, 2003.

[19] Y. Sankarasubramaniam, O. B. Akan, I. F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," in *Proc. ACM MOBIHOC 2003*, pp.177-188, Annapolis, Maryland, USA, June 2003.



### 이 좌 형

e-mail : jihlee@snsnlab.kangwon.ac.kr

2003년 강원대학교 정보통신공학과 (공학사)

2005년 강원대학교 컴퓨터정보통신공학과 (공학석사)

2005년~현 재 강원대학교 컴퓨터정보통신 공학과(박사과정)

관심분야: 멀티미디어 시스템, 센서 네트워크



### 선 주 호

e-mail : jihseon@snsnlab.kangwon.ac.kr

2002년~현 재 강원대학교 컴퓨터정보통신 공학과 재학

관심분야 : 센서네트워크, Ad-Hoc, 임베디드 시스템



### 정 인 범

e-mail : ibjung@snsnlab.kangwon.ac.kr

1985년 고려대학교 전자공학과 학사.

1985년~1995년 (주) 삼성전자 컴퓨터 시스템사업부 선임 연구원

1992년~1994년 한국과학기술원 정보통신공학과(석사)

1995년~2000년 8월 한국과학기술원 전산학과(박사)

2001년~현 재 강원대학교 컴퓨터정보통신공학전공 교수

관심분야 : 운영체제, 소프트웨어 공학, 멀티미디어 시스템, 센서네트워크