

## 전송지연요소 회피를 통한 실시간 센서 네트워크 라우팅

신수정<sup>o</sup> 차호정  
연세대학교 컴퓨터과학과  
{acid<sup>o</sup>, hjcha}@cs.yonsei.ac.kr

Towards Real-time Routing via Avoiding Transmission Delay in Wireless Sensor Networks

Sujeong Shin<sup>o</sup> Hojung Cha  
Dept. of Computer Science, Yonsei University

### 요 약

센서네트워크에서 이동물체의 경로 추적과 같이 실시간으로 패킷이 목적노드에 도착하여야 데이터가 유효한 응용의 경우에는 패킷 전송의 실시간을 보장할 수 있는 전송경로설정이 필요하다. 본 논문에서는 패킷을 전송하는데 있어서 지연시간을 증가시키는 요소인 혼잡상황과 낮은 링크 퀄리티를 가지는 경로를 회피함으로써 전송지연시간을 최소화 하는 전송경로 설정 기법을 제안한다. 네트워크의 위상에 따라 노드마다 다른 양상을 보이는 채널 부하도를 효과적으로 측정하고 네트워크의 혼잡상황을 판단하는 기준으로 삼는다. 이와 함께 링크의 전송 성공률을 이용하여 이웃 노드들의 전송 경로 선택의 적합도를 계산하고 확률적으로 경로를 선택한다.

### 1. 서 론

무선 센서네트워크의 응용 중 물체의 이동경로 추적이나 재해 상황의 감지 등의 경우는 데이터가 빠른 시간 내에 사용자에게 전달되어야 한다. 이러한 응용들을 위해서는 실시간 데이터 전송을 보장할 수 있는 전송 기술을 지원해야 한다. 보통 노드들이 넓은 지역에 분포되어 있고, 통신 가능한 전파 거리의 범위가 한정되어있기 때문에 데이터를 생성하는 소스노드에서 목적노드까지의 데이터 전송은 일반적으로 멀티 홉 방식을 사용한다. 기존 연구에서는 목적노드까지 전송 경로가 가장 짧고, 홉카운트가 적은 전송경로를 이용하였다[1]. 그러나 최소 전송 경로가 최소 지연 시간을 보장할 수는 없는데 가장 큰 원인으로 네트워크의 혼잡 상황을 들 수가 있다. 센서 네트워크의 특성상 다수의 소스 노드에서 하나 혹은 소수의 싱크 노드로 데이터를 전송하는 것이 일반적이므로 싱크 노드에 가까운 노드나 데이터가 비교적 많이 생성되는 지역에서는 패킷이 물리게 되고 이는 네트워크의 혼잡 상황을 일으키는데 채널 경쟁이나 전송 실패로 인해 전송 지연 시간이 크게 증가하게 된다.

전송 실패로 인한 전송 지연 시간의 증가는 혼잡 상황 이외에도, 라디오 링크의 특성에 기인한다. 패킷을 송수신하는 노드 사이의 거리가 멀어질수록, 패킷 수신 성공률이 떨어질 뿐 아니라, 성공률의 편차가 큰 양상을 보인다[2]. 따라서 데이터 전송 지연시간을 줄이기 위해서는 이러한 점을 고려해야 한다.

과거 연구에서 실시간 전송 경로를 설정하는 연구는 패킷의 전송제한시간 및 소스노드에서 싱크노드까지의 거리를 기준으로 패킷 스케줄링을 하고 노드와 노드간의 전송 지연시간을 가지고 라우팅 경로를 설정하는데 초점이 맞춰져있다. SPEED[3]에서는 지리적인 위치정보를 이용하여 전송경로를 설정하되 Speed (거리/시간) 개념을 도입하여 싱크노드까지의 거리가 짧으면서 노드 간 전송 지연시간이 적은 이웃노드가 좀 더 높은 확률로 패킷을 전달할 수 있게 한다. Seada[4]는 라디오 링크

의 특성을 고려하여 거리와 링크의 패킷 수신 성공률의 곱이 클수록 높은 확률로 전송경로가 설정될 수 있게 하는 기법을 제시한다. 한편 패킷의 전송지연시간을 증가시키는 요소 중 하나인 네트워크의 지역적 혼잡상황을 감지하기 위해 Shin[5]은 채널부하도를 이용하여 노드마다 동적으로 혼잡감지 기준값을 계산함으로써 센서 네트워크 특성에 맞는 혼잡 감지기법을 제시하였다.

본 논문에서는 관찰된 사실에 바탕하여 실시간성 데이터 전송을 위한 새로운 라우팅 기법을 제안하고자 한다. 제안하는 라우팅 기법은 패킷의 전송 지연시간을 증가시키는 요소를 분석하여 이를 회피할 수 있는 경로를 확률적으로 회피하는 정책을 사용한다.

### 2. 전송지연요소 회피 정책을 이용한 실시간 라우팅

본 논문에서 제안하는 라우팅 알고리즘의 목표는 패킷의 전송 지연 시간을 될 수 있는 한 최대한으로 줄이는 데에 있다. 기본적으로 네트워크에 존재하는 모든 노드들은 이웃노드의 정보와 싱크노드의 위치정보를 이용하여 경로를 설정하는 분산화된 Geographic 전송경로 설정 알고리즘을 사용한다. 그리고 소스노드에서 싱크 노드까지 패킷을 전송하는데 최적의 전송경로 하나만을 이용하는 것이 아니라, 전송경로로 사용할 수 있는 모든 이웃노드들의 적합성을 제안하는 가중함수를 이용하여 계산한 후 확률적으로 매 전송 시 다른 경로를 선택할 수 있는 비결정적 방식을 사용한다. 이 방식은 네트워크의 트래픽 밸런싱을 맞추어 네트워크 분리 등의 위험을 방지할 수 있는 장점이 있다.

제안하는 라우팅 알고리즘은 패킷의 전송지연 시간을 줄이기 위해서 소스노드에서 싱크노드까지 최단 경로 및 최소 홉카운트를 유지하되 전송 지연시간을 증가시키는 요소를 회피하는 정책을 사용한다. 앞서 언급했듯이 전송 지연 시간을 증가시키는 가장 큰 원인은 네트워크의 혼잡상황을 들 수 있다. 혼잡 상황이 발생했을 시 노드 내에서 전송 큐에서의 대기시간, 채널 사용 경쟁을 위한 Back-off 시간, terminal node problem

본 연구는 과학기술부에서 지원하는 국가지정연구실사업으로 수행하였음 (과제번호 : 2005-01352)

등의 문제로 전송 실패 시 재전송 정책 등의 지연시간이 발생한다. 따라서 전송 경로를 설정할 때 지역적인 혼잡 상황이 발생했을 때는 이 지역은 회피하여야 한다. 혼잡 상황을 감지하기 위해서 채널 부하도를 이용한다. 주기적으로 채널의 상태를 측정하여 이에 대한 평균값을 낸 것을 채널부하도라 하는데 지난 연구[5]에서는 센서 네트워크 위상에 따라 노드들마다 다른 채널부하도 양상을 보이는 것을 관찰하여 이에 적절한 채널 혼잡 감지 기법을 제시하였다. 본 논문에서는 [5]에서 제시한 채널 혼잡 감지 기법을 사용한다.

먼저 주기적으로 식 (1)과 같이 채널 부하도  $\bar{\Phi}_n$ 를 측정한다.  $\bar{\Phi}_0$ 는 특정 채널 상태가 유지되기 시작할 때의 채널 부하도를 나타내고,  $n$ 은 채널 상태가 유지되는 기간의 샘플링 횟수를 의미한다.

$$\bar{\Phi}_n = \begin{cases} (1-\alpha)^n \bar{\Phi}_0 & (\text{idle 상태}) \\ (1-\alpha)^n (\bar{\Phi}_0 - 1) + 1 & (\text{busy 상태}) \end{cases} \quad (1)$$

채널부하도  $\bar{\Phi}_n$ 를 계산한 이후에, 채널부하도의 양상을 파악하기 위해 두 개의 FIFO Queue를 이용하여 채널부하도의 상하한선을 계산한다. 하나의 큐에는 busy-idle로의 상태 변화 시점에서의 채널부하도 값을 저장하고, 다른 하나에는 idle-busy로의 상태 변화 시점에서의 채널부하도 값을 저장한다. 그리고 식 (2)에서 busy-idle 큐의 최대값의 지수가중이동평균값으로 상한선  $C_{upper}$ 를 계산하고, idle-busy 큐의 최소값의 지수가중이동평균값으로 하한선  $C_{lower}$ 를 계산한다.

$$\begin{aligned} C_{upper, i+1} &= (1-\beta)C_{upper, i} + \alpha \max(\text{Queue}_{\text{busy-idle}}) \\ C_{lower, i+1} &= (1-\beta)C_{lower, i} + \alpha \min(\text{Queue}_{\text{idle-busy}}) \end{aligned} \quad (2)$$

지연시간을 증가시키는 또 다른 요소로 낮은 링크 퀄리티로 인한 전송 실패를 들 수 있다. 따라서 제안하는 전송 경로 설정 기법은 모든 이웃노드들에 대해 주기적으로 패킷 수신 성공률을 측정하고 EWMA 평균값을 계산하여 경로설정의 적합성을 판단하는데 사용한다.

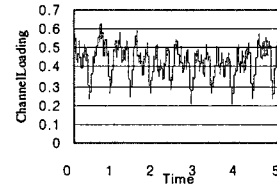
모든 노드들은 주기적으로 이웃노드들에게 자신의 정보를 담은 Beacon 메시지를 브로드캐스트한다. 이 정보를 이용하여 패킷을 전송하고자 하는 노드  $Node_i$ 는 모든 이웃노드  $Node_j$ 에 대해서 식(3)와 같이 가중치를 계산하여 확률적으로 다음 노드를 선택한다. 전송 지연 시간이 적게 소요될 수 있는 이웃노드가 가장 높은 확률로 선택될 수 있게 하는 것이 식(3)의 목표이다.

$$\text{Weight}(Node_j) = \frac{\{Dist(Node_i) - Dist(Node_j)\} \times R_{rec}(Node_j)}{Load(Node_j)} \quad (3)$$

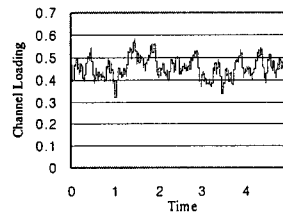
$\text{Weight}(Node_j)$ 는 이웃노드  $Node_j$ 의 가중치이고,  $Dist(Node)$ 는 노드에서 싱크노드까지의 거리이다.  $R_{rec}(Node_j)$ 는 현재 노드  $Node_i$ 와 이웃노드  $Node_j$  간 링크의 패킷 수신 성공률을 의미하고,  $Load(Node_j)$ 는 이웃노드에 걸린 채널 부하도이다.  $Load(Node_j)$ 는 현재의 채널부하도를 상한선 값의 상대적인 비율로 계산하여 0에서 1사이로 값을 결정한다. 모든 이웃노드



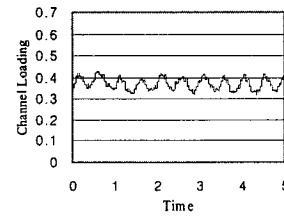
그림 1 실험환경



a) sampling rate = 200us



b) sampling rate = 400 us



c) sampling rate = 1ms

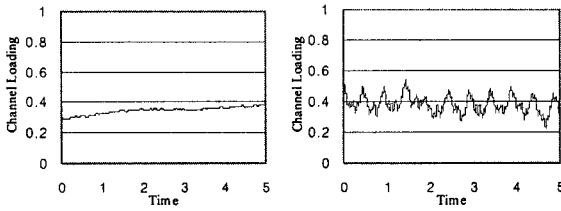
그림 2 Sampling Rate 에 따른 채널부하도

들에 대해 식(3)를 이용하여 가중치를 계산하고 가중치가 0보다 큰 이웃노드들의 모든 가중치의 합이 1이 되도록 정규화한다. 만약 모든 이웃노드들이 가중치가 0보다 작은 경우는 GPSR[1]에서 제시한 Void problem에 빠져있는 경우로 해석할 수 있다. 이 때 랜덤하게 경로를 선택할 경우에는 싱크노드까지 패킷을 전송하지 못하고 루프에 빠질 수 있으므로, GPSR[1]에서 제안한 Right-Hand 방식대로 가중치가 0이상인 이웃노드가 존재하는 노드를 만날 때까지 반시계방향에 위치한 노드에게 패킷을 전달하여 이 문제를 해결한다.

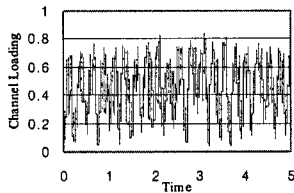
### 3. 실험

본 논문에서 제안하는 라우팅 알고리즘을 검증하기 위하여, 제안하는 알고리즘과 GPSR[1], SPEED[3]을 구현하여 실험을 수행하고 성능을 비교 평가하였다. 구현에 사용한 센서 노드는 U.C. Berkeley에서 개발한 Telos(rev.b)이다. 운영체제는 TinyOS[6]를 사용하였다. 그림 1과 같이 실험에 사용된 노드는 49개로 7x7 grid 형태로 배치하였다. 노드간의 간격은 실험의 편의를 위해 1.5m로 제한하였고, 노드의 라디오 범위는 6m 이내로 조절하였다. 실험에서 노드들이 전송하는 패킷의 크기는 32B이고 노드의 전송률은 250Kbps이다.

먼저 채널 부하도 측정에 필요한 파라미터를 결정하기 위한 실험을 진행하였다. Sampling Rate를 {100us, 400us, 1ms}로 설



a)  $\alpha=0.0001$                       b)  $\alpha=0.001$



c)  $\alpha=0.01$

그림 3 EWMA 가중치에 따른 채널부하도

정하여 채널부하도를 측정하였다. 그림 2.b)에서 보듯 sampling rate 를 400us 로 설정했을 때 채널부하도를 적절하게 측정하는 것으로 판단하였다.

다음으로 sampling rate 를 400us 로 고정하고 EWMA 가중치  $\alpha$ 를 0.0001, 0.001, 0.01 로 변화시켰을 때 채널부하도의 변화를 관찰하였다.(그림 3) 가중치를 적게 둘 경우 현재의 채널상태가 전체 채널부하도의 값을 결정하는데 매우 적은 영향을 미쳐 채널부하도의 값의 변화가 거의 없는 모습을 살펴볼 수 있다. 반대로 가중치를 크게 둘 경우 현재의 채널 상태로 인해 채널부하도가 빈번하게 극단적으로 변하는 모습을 관찰할 수 있다. 본 실험에서는 그림 3.b)과 같이 가중치를 0.01 로 설정하였다. 마찬가지로 상하한선에 필요한 EWMA 가중치  $\beta$ 에 대해 같은 실험을 수행하여  $\beta$ 값은 0.01로 결정하였다.

정사각형 Grid 형태로 배치된 49개의 노드들 중에서 CBR traffic(4 packets/sec)을 발생시키는 노드를 4개에서 20개까지 증가시켜 End-to-End 전송 지연시간과 패킷 도착 성공률의 평균값을 비교하였다. CBR traffic 을 발생시키는 노드들은 topology 상에서 고르게 분산 배치 시켰다. 그림 4과 5에서 GPSR는 전송 지연시간과 패킷 도착 성공률 면에서 가장 안 좋은 성능을 보였다. 싱크 노드 주변의 노드에게 네트워크 트래픽이 몰리는데 홉카운트가 가장 적은 경로 하나만을 이용하므로 이 경로에 속하는 노드들에 채널경쟁이 심해져 이로 인한 지연시간이 증가한다. 또한 링크클리티어를 고려하지 않았으므로 낮은 링크 수신률에 의한 전송실패 때문에 평균 패킷 도착 성공률이 떨어지는 것을 알 수 있다. 노드 간 패킷 전송 지연시간과 목적노드까지의 거리를 기준으로 전송경로를 설정하는 SPEED의 경우 GPSR 보다 좋은 성능을 보였지만 본 논문에서 제안하는 알고리즘은 SPEED에 비해 14% 정도의 빠른 Delay Time 성능을 보였다. Delivery Success Ratio 면에서도 본 논문에서 제안하는 알고리즘이 가장 좋은 성능을 보였다. 낮은 Link Quality 와 혼잡 상황 발생으로 인해 패킷이 손실되는 것을 회피하려는 정책이 성공했기 때문으로 판단된다.

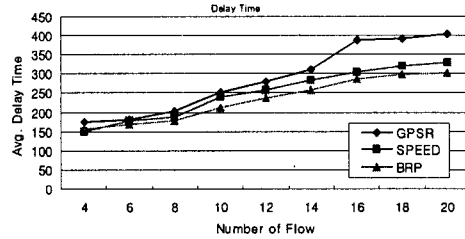


그림 4 End-to-End Delay Time

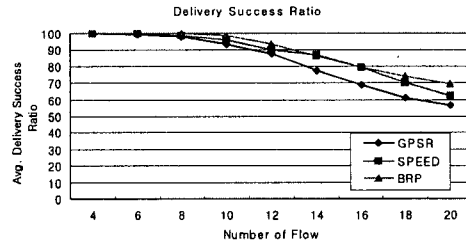


그림 5 End-to-End Delivery Success Ratio

#### 4. 결론

본 논문에서는 end-to-end 의 전송지연시간을 최소로 줄이기 위해 전송 지연시간을 발생시킬 수 있는 요소를 분석하고 이를 회피하는 정책을 사용하는 geographic 라우팅 알고리즘을 제안 하였다. 노드들의 위치 정보를 이용하여 전송 경로로 설정 가능한 이웃노드들 중에서 전송 시간을 지연시킬 수 있는 혼잡지역에 속한 노드나 link quality 가 낮은 노드는 전송경로로 선택될 수 있는 확률을 낮게 하여 확률적으로 전송경로를 선택한다. 채널의 혼잡상황을 판단하는데 있어서 채널부하도를 이용하여 동적으로 혼잡상황 감지 기준값을 계산하여 적용한다. 제안한 기법은 실제 Telos 하드웨어를 가지고 TinyOS 운영체제 상에서 구현을 하여 성능을 평가하였다. 전송 지연시간 과 패킷 도착 성공률 면에서 좋은 성능을 나타내는 것을 보였다.

#### 5. 참고문헌

- [1] B. Karp, and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *In Proc. of MobiCom 2000*, August, 2000.
- [2] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," *In Proc. of International Conference on Embedded networked sensor systems*, Los Angeles, California, USA, November, 2003
- [3] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," *In Proc. of International Conference on Distributed Computing Systems (ICDCS 2003)*, Providence, RI, May, 2003.
- [4] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks," *In Proc. of Sensys*, Baltimore, Maryland, USA, November, 2004.
- [5] 신수정, 차호정, "채널부하 측정에 기반한 센서네트워크 혼잡 감지 기법," *In Proc. of KCC 2005*, Korea, July, 2005.
- [6] <http://www.tinyos.net>