

데이터 퓨전을 위한 무선 센서 네트워크용 스패닝 트리 : FST

서 창 진[†] · 신 지 수^{**}

요 약

무선 센서 네트워크(Wireless Sensor Network : WSN)는 자율적으로 라우팅 경로를 구성하여 물리적으로 떨어진 지역의 데이터를 수집하는 무선망이다. 본 논문은 주기적으로 수집한 모든 데이터를 하나의 기지 노드로 전달하는 convergecast 환경에서 퓨전(fusion)을 반영한 라우팅 방법을 제안한다. 지금까지 대부분의 연구는 무퓨전(no-fusion)과 전퓨전(full-fusion)의 두 경우만을 다루었다.

제안하는 Fusion rate based Spanning Tree(FST)는 임의의 퓨전율 f ($0 \leq f \leq 1$)에서 총 전송 에너지 비용을 줄이는 라우팅 경로를 제공한다. FST는 무퓨전($f = 0$)과 전퓨전($f = 1$)의 convergecast에서 각각 최적의 토폴로지인 최소 경로 트리(Shortest Path spanning Tree : SPT)와 최소 스패닝 트리(Minimum Spanning Tree : MST)를 제공하며, 임의의 f ($0 < f < 1$)에 대해서도 SPT나 MST보다 우수한 토폴로지를 제공한다. 시뮬레이션은 100-노드 WSN에서 모든 f ($0 \leq f \leq 1$)에 대해 FST의 총 길이가 평균적으로 MST보다 약 31%, SPT보다 약 8% 절약을 보여준다. 따라서 우리는 FST가 WSN에서 매우 유용한 토폴로지임을 확인하였다.

키워드 : 퓨전, 센서 네트워크, 스패닝 트리, 라우팅, FST

FST : Fusion Rate Based Spanning Tree for Wireless Sensor Networks

Changjin Suh[†] · Jisoo Shin^{**}

ABSTRACT

Wireless Sensor Network (WSN) is a wireless network that gathers information from remote area with autonomously configured routing path. We propose a fusion based routing for a 'convergecast' in which all sensors periodically forward collected data to a base station. Previous researches dealt with only full-fusion or no-fusion case.

Our Fusion rate based Spanning Tree (FST) can provide effective routing topology in terms of total cost according to all ranges of fusion rate f ($0 \leq f \leq 1$). FST is optimum for convergecast in case of no-fusion ($f = 0$) and full-fusion ($f = 1$) and outperforms the Shortest Path spanning Tree (SPT) or Minimum Spanning Tree (MST) for any range of f ($0 < f < 1$). Simulation of 100-node WSN shows that the total length of FST is shorter than MST and SPT nearby 31% and 8% respectively in terms of topology lengths for all range of f . As a result, we confirmed that FST is a very useful WSN topology.

Keywords : Fusion, Sensor network, Spanning tree, Routing, FST

1. 서 론

WSN은 한정된 에너지를 가진 노드들이 자율적으로 망을 구성하여 원거리 지역의 다양한 상태 정보를 수집하는 망이다.[1][2] WSN 라우팅에서 매우 유용하게 사용하는 convergecast[3][4]는 방송과는 역방향의 통신으로 수집한 모든 정보를 기지 노드(Base Station : BS)라 부르는 하나

의 노드로 전송한다. convergecast가 효율적으로 수행된다면 모든 노드는 자신의 이웃 노드로 단 한 번의 데이터만 전송하여 convergecast를 완료할 수 있다. 그 결과 라우팅 경로는 하나의 스패닝 트리를 생성한다. Convergecast는 라우팅을 수행하는 동안 퓨전이라 부르는 데이터 축약이 발생하여 전달하는 데이터의 양을 줄인다. 최근까지 대부분의 연구들은 무퓨전과 전퓨전의 두 경우만을 제한적으로 다루었다.

우리는 퓨전율 f 를 각 센서 노드가 이웃으로부터 수신한 데이터와 자신이 생성한 데이터를 병합하여 다음 노드로 전달할 때 전송할 데이터의 크기가 줄어드는 비율로 정의한다.[5] 전퓨전에서는 노드가 여러 개의 x -비트 패킷을 수신한 후 한 개의 x -비트 패킷으로 축약하여 전송한다. 반면

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

† 종신회원 : 숭실대학교 컴퓨터학부 교수

** 정 회 원 : 숭실대학교 컴퓨터학과 박사과정

논문접수 : 2008년 9월 17일

수정일 : 1차 2008년 12월 10일

심사완료 : 2008년 12월 11일

무퓨전에서는 전송 패킷의 길이가 수신한 전체 패킷의 길이와 자신이 생성한 패킷 길이의 합과 같다.

Convergecast에 유용한 대표적인 토폴로지로 최소 스패닝 트리(Minimum Spanning Tree : MST)와 최단 경로 스패닝 트리(Shortest Path spanning Tree : SPT)가 있다. MST는 모든 노드를 연결하는 링크의 총길이를 최소로 만들고, SPT는 루트 노드에서 모든 노드로 최소 경로를 제공한다. 만일 우리가 MST와 SPT를 데이터 퓨전의 관점에서 본다면 MST는 전퓨전일 경우에, SPT는 무퓨전일 경우에 데이터 전송에 필요한 에너지를 최소로 사용한다. 본 논문은 퓨전 기반 convergecast의 연구로 Fusion rate based Spanning Tree(FST)를 제안한다. FST는 무퓨전일 경우 SPT와 동일하고, 전퓨전일 경우 MST와 동일하여 최적치를 유지하며 그 이외의 경우에도 항상 SPT와 MST보다 우월하다.

지금부터는 퓨전과 관련된 스패닝 트리에 대한 기존 연구를 살펴본다. [6]은 convergecast에서 에너지와 더불어 동작하는 '시간'을 최소로 사용하는 스패닝 트리를 찾는다. 이 논문에서 시간은 무선 TDMA 전송에서 convergecast에 필요한 시간 슬롯의 수를 의미한다. 이 논문은 자신들이 정의한 'time performance index'를 고려하여 동작 시간을 줄이는 스패닝 트리를 제안한다. 그 결과 제안하는 알고리즘은 균형 트리에 가까운 경로를 구성한다.

[7]과 [8]은 각각 Minimum Fusion Steiner Tree(MFST)와 Binary Fusion Steiner Tree(BFST)를 제안하였다. MFST와 BFST는 센서 노드가 높은 연산 능력을 가지며 각 노드는 퓨전하는 동안 많은 에너지를 소비한다는 가정 하에서 전송 에너지와 퓨전 에너지의 합을 최소화하려 했다. MFST에서는 모든 퓨전이 중간 노드에서 발생한다고 가정했다. MFST의 분석에 따르면 퓨전율이 낮거나 퓨전 에너지가 많이 소비될 경우에는 데이터 전송보다 퓨전에 의해 더 많은 에너지를 소비하는 경우가 발생한다. 따라서 BFST는 퓨전을 수행하기 전에 퓨전 가능성 시험을 수행하고 소비 에너지를 절약할 수 있는 경우에만 퓨전을 수행한다. MFST와 BFST는 패킷 전송에 관한 가정을 매우 간소화하여 전송 거리에 관계없이 동일한 전송 에너지를 소비한다고 가정하고 문제를 풀었다. 저자가 살펴본 바에 따르면 본 논문과 같이 토폴로지와 퓨전을 동시에 고려하는 관련 연구는 찾을 수 없었다.

본 논문에서는 다음의 세 가지 아이디어를 제안한다. 첫째, 퓨전을 고려하여 토폴로지 길이를 정의하는 방법과 둘째, 항상 퓨전을 수행하는 일반적인 퓨전 모델과 달리 one-time 퓨전 모델을 제안한다. 마지막으로 FST를 제안한다.

본 논문은 다음과 같이 구성된다. 우리는 2장에서 여러 종류의 거리를 우선 정의하고 이 정의를 이용하여 MST와 FST, SPT를 정의한다. 3장에서는 우리가 제안하는 FST를 만드는 방법과 성질에 대해서 설명한다. 4장에서는 시뮬레이션을 통해 FST의 총길이를 SPT와 MST에 대해서 비교하고 5장에서 결론을 맺는다.

2. 거리 및 문제 정의

우리는 우선 거리를 새롭게 정의한다. 2.1에서는 퓨전을 f 가 반영되는 거리인 $d(\cdot)$ 를 포함한 세 가지의 거리를 정의한다. $d(\cdot)$ 를 사용하면 무퓨전과 전퓨전 모델을 포함하여 모든 퓨전 모델을 하나의 f -퓨전 모델 ($0 \leq f \leq 1$)로 통합할 수 있으며 FST와 MST, SPT 문제를 하나의 문제로 정의할 수 있다.

2.1 거리의 정의

본 절에서는 여러 가지 용어를 정의한다. $l(i,j)$ 는 노드 i 에서 노드 j 로 향하는 링크이고 그 링크 비용을 $C(i,j)$ 라 한다. 노드 q 의 k 번째 부모 노드를 $p_k(q)$ 라 한다. 즉, $p_1(q)$ 는 q 의 부모 노드이며 $p_0(q)$ 는 q 자신이다. 우리는 convergecast의 목적지 노드를 루트 노드로 정의하고 R 로 표시한다. 편의를 위해 루트 노드 R 의 부모 노드는 자신이라 가정한다. 따라서 식 (1)을 얻을 수 있다.

$$p_k(R) = R, \quad k=0,1,2,\dots \quad (1)$$

노드 q 의 거리는 스패닝 트리 상에서 q 부터 R 까지의 거리로 정의할 수 있다. 그러나 퓨전을 고려하면 거리의 정의가 모호해지기 때문에 이를 극복하기 위해서 새로운 거리 규칙이 필요하다.

우리는 세 가지 거리 규칙, 즉, i) 퓨전을 고려하지 않는 일반적인 거리 $\delta(\cdot)$ 와 ii) 일정한 퓨전율을 사용하는 거리 $\Lambda(\cdot)$, iii) one-time 퓨전 거리 $d(\cdot)$ 를 제안한다. 이후부터는 노드 q 를 R 의 k 번째 ($k=1,2,\dots$) 후손 노드로 사용한다. $\delta(\cdot)$ 는 퓨전을 고려하지 않고 단순히 모든 링크 비용을 합한 값으로서 식 (2)와 같다.

$$\delta(q) = \sum_{i=1,k} C_{p_{i-1}(q),p_i(q)}. \quad (2)$$

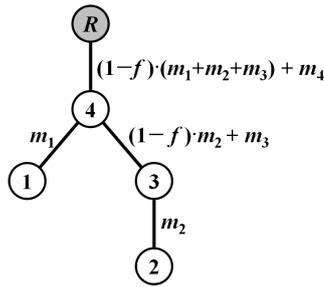
$\Lambda(\cdot)$ 는 일반적으로 제안되는 퓨전 모델로 [6]에서도 동일한 방법이 언급되었다. $\Lambda(\cdot)$ 는 모든 중간 노드를 거칠 때마다 데이터가 일정한 비율 f 로 축약된다는 가정 하에 정의되는 거리로서 식 (3)으로 나타낼 수 있다.

$$\Lambda(q) = \sum_{i=1,k} (1-f)^{i-1} \cdot C_{p_{i-1}(q),p_i(q)} \quad (3)$$

마지막으로 우리가 제안하는 one-time 퓨전에서 적용되는 거리 규칙 $d(\cdot)$ 는 다음과 같다.

$$d(q) = \begin{cases} C_{q,p_1(q)} & : p_1(q) = R \\ C_{q,p_1(q)} + (1-f) \cdot \sum_{i=2,k} C_{p_{i-1}(q),p_i(q)} & : p_1(q) \neq R \end{cases} \quad (4)$$

$d(q)$ 로 정의한 one-time 퓨전은 이름에서 암시하듯이 부모 노드에서 한 번만 퓨전이 발생한다. (그림 1)은 노드 q 에서 m_q 길이의 데이터가 생성되어 스패닝 트리를 따라 R 로 전달될 때 각 링크 $l(i,j)$ 에서의 데이터 길이를 보여준다. 우리는 모든 소스 노드가 동일한 길이의 데이터를 생성한다고 가정하고 있지만 독자들의 이해를 돕기 위해서 (그림 1)에서는 메시지가 발생한 소스 노드를 언급하였다.



(그림 1) 퓨전의 예 : 거리 규칙 $d(\cdot)$ 를 사용할 때 전송 데이터의 길이

지금부터는 $d(\cdot)$ 를 사용하여 스페닝 트리 T_S 의 총길이 $L(T_S)$ 를 계산한다. $L(T_S)$ 는 T_S 에서 루트 노드 R 로부터 모든 노드 q 까지의 거리의 합으로 정의한다.

$$L(T_S) = \sum_{\{q|q \in T_S\}} d(q). \tag{5}$$

$L(T_S)$ 를 최소화하는 문제는 식 (4)에서 퓨전을 f 에 0을 할당하면 SPT 문제와 같아지고 1을 할당하면 MST 문제와 동일해진다.

2.2. 문제 정의

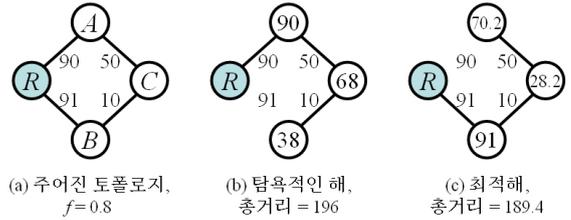
우리는 우선 환경을 가정하고 문제를 정의한다.

- WSN과 convergecast에 대한 가정 : 모든 노드의 위치는 알려져 있고 모든 노드는 주기적으로 고정된 길이의 데이터를 발생시켜 루트 노드 R 로 convergecast한다. 또한 우리는 데이터 퓨전이 적용되어 각 링크에 어떤 크기의 데이터가 전송되었는지에 대한 충분한 통계를 가지고 있다고 가정한다.
- 문제의 정의 : WSN과 convergecast에 대한 가정 하에서 convergecast에 소비되는 전송 에너지의 합을 최소화하는 스페닝 트리 T_S 를 찾는다. 수학적으로 다시 말하면 WSN과 convergecast에 대한 가정 하에서 식 (4)와 (5)를 사용하여 총길이가 최소인 토폴로지 T_S 를 찾는다.

2.3 새로운 거리 규칙의 분석

거리 규칙 $d(\cdot)$ 와 $\Lambda(\cdot)$ 는 링크 비용 $C(i,j)$ 가 주어진 망에서 비선형적인 성질을 가지기 때문에 심지어 후손 노드들이 부모 노드들보다 더 짧은 거리를 가지는 경우도 발생할 수 있다. $d(\cdot)$ 를 사용할 경우 R 에 가까이 위치한 노드들은 짧은 링크를 선호하지만 멀리 위치한 노드들은 경로가 다소 길어도 R 로 향하는 직선 경로를 선호한다. 반면 $\Lambda(\cdot)$ 를 사용할 경우 대부분의 중간 노드들은 짧은 링크를 선호한다.

이 논문의 문제는 비탐욕적인 방법으로 풀어야 한다. (그림 2)는 탐욕적인 해와 최적해가 불일치하는 일례를 제시한다. (그림 2.(a))는 링크값이 주어진 4-노드 망을 보여준다. 이 토폴로지에 탐욕적인 방법을 적용하면 루트 노드 R 에서 최초의 링크를 선택할 때 두 개의 인접 링크 중에 거리가 짧은 (R,A) 를 선택한다. (R,A) 를 선택한 상태에서 최적한



(그림 2) 탐욕적인 해와 최적해가 다른 경우

해가 (그림 2.(b))에 표시되어 있다. 이 토폴로지의 각 노드에서의 길이는 노드를 표시하는 원 안에 표시되어 있으며 이를 모두 합한 총길이는 196이다. (그림 2.(a))의 최적해는 (그림 2.(c))에 제시되었는데 이 토폴로지의 총길이는 189.4로 (그림 2.(b))보다 6.6이 짧다. 또한 (그림 2.(b))와 (그림 2.(c))에서는 후손 노드가 자신의 부모 노드보다 거리가 짧은 경우가 발생할 수 있다. 이러한 비선형적인 특징으로 인해서 우리는 수학적 최적 해법 또한 존재하지 않는다고 믿는다. 우리의 해법은 이러한 간단한 예외적인 상황을 감지하여 최적해를 제시할 수 있다.

일정한 퓨전율을 사용하는 $\Lambda(\cdot)$ 보다 one-time 퓨전 $d(\cdot)$ 를 사용하는 이유는 다음과 같다. 첫째, one-time 퓨전은 간단하다. 전달하는 데이터의 길이는 (그림 1)과 식 (11)과 같이 항상 f 에 대해 1차 방정식으로 추역된다. 이러한 성질을 이용하여 우리는 통계 결과에 기초하여 적절한 퓨전율 f 를 수학적으로 추출할 수 있다. 둘째, 일정한 퓨전율을 사용하면 0이 아닌 값에 수렴할 수 없다. 일반적으로 퓨전을 반복할수록 퓨전율은 줄어들고 데이터의 길이는 퓨전이 무한히 발생하더라도 특정한 값에 수렴한다. 만일 $\Lambda(\cdot)$ 를 사용할 경우 충분한 숫자의 중간 노드를 거치면 데이터 길이는 항상 0으로 추역된다. 그 결과 $\Lambda(\cdot)$ 를 사용하면 여러 홉으로 전달되는 우회경로를 선호하는 모순이 발생한다.

3. FST

이 장에서는 FST의 동작을 두 단계로 정의한다. 단계 1에서는 통계를 사용하여 최적의 퓨전율 f_0 를 계산한다. f_0 는 주어진 스페닝 트리 T_S 에서 계산된 최적의 f 값이다. 단계 2에서는 f_0 를 사용하여 스페닝 트리 T_S 를 생성한다. 우선 단계 2-1에서 초기 스페닝 트리를 생성한 후에 단계 2-2에서 재구성을 통해 스페닝 트리를 더 짧게 변형한다.

3.1 단계 1 : 최적 퓨전율의 계산

본 논문의 주목적이 최적의 스페닝 트리를 찾는 일이므로 단계 1은 부록에서 언급한다.

3.2 단계 2 : 스페닝 트리 생성

우리는 경험적 기법을 통해 FST를 두 단계로 생성한다. 초기 트리는 단계 2-1에서 하향방향(루트 R 에서 리프 노드로)으로 생성하고 단계 2-2에서 더 짧은 토폴로지를 찾기 위해서 상향방향(리프 노드에서 루트 R 로)으로 검색하며 개

선할 토폴로지가 있다면 재구성한다.

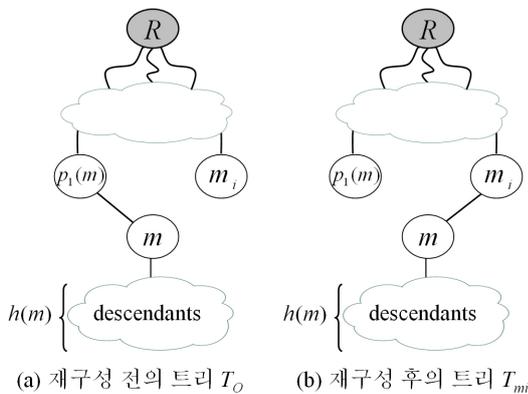
3.2.1 단계 2-1 : 초기 트리 생성

우선 초기 트리를 생성하는 방법을 정의하고 단계 2-1의 규칙을 설명한다. 초기 트리는 Dijkstra의 SPT 알고리즘과 Prim의 MST 알고리즘을 혼용하여 생성한다. 우선 Dijkstra의 SPT 알고리즘을 사용하여 루트 노드 R 에서 시작하여 R 을 포함한 n_S 개의 노드를 선택할 때까지 트리를 부분적으로 생성한다. 그 다음 Prim의 MST 알고리즘을 사용하여 선택하지 않은 노드들을 추가하면서 n -노드 스페닝 트리를 완성한다. n_S 는 식 (6)으로 계산한다. $\lceil x \rceil$ 는 x 보다 같거나 크며 x 와 가장 가까운 정수이다. 우리는 식 (6)에서 다양한 값의 n -노드 WSN과 f 값에 대해서 시뮬레이션한 결과 W 를 8로 고정하였다.

$$n_S = \lceil (1-f^W) \cdot n \rceil, \quad W=8. \quad (6)$$

이 단계에서 우리는 다음의 두 가지 특징을 찾을 수 있다. 첫째, 식 (6)에서 W 가 8이므로 대부분의 퓨전을 f 에 대해 n_S 는 n 에 매우 근접한다. 이는 초기 FST가 SPT와 매우 유사함을 뜻한다. 둘째, R 에서 가까운 노드는 이웃 노드를 선택할 때 SPT 알고리즘을 선호하며 먼 노드는 MST 방식을 선호한다. 이는 많은 후손 노드를 가진 노드가 이웃 노드를 선택할 때 SPT 규칙을 선호하기 때문이다. 그 이유를 다음에서 설명하자. (그림 3)에서 노드 m 이 두 개의 후보 노드 중에서 자신의 부모 노드를 선택해야 하고 두 후보 노드가 다음과 같은 특성을 가진다고 생각해 보자. 첫째는 후보 노드까지의 비용은 짧지만 후보 노드에서 R 까지의 거리가 긴 경우이고, 둘째는 후보 노드까지의 비용은 길지만 후보 노드에서 R 까지의 거리가 짧은 경우이다. R 로 향하는 첫 번째 홉의 링크 비용은 $d(m)$ 의 100%이지만 이후의 홉에 대해서는 $d(m)$ 의 $(1 - f)$ 의 비율로 줄어든다. 노드 m 이 리프 노드일 경우 R 로 향하는 첫 번째 홉이 이후의 홉보다 중요하기 때문에 전자를 선호한다. 반면 노드 m 이 많은 후손 노드를 가졌다면 후자를 선호한다. 이는 많은 m 의 후손 노드들이 짧은 경로를 선호하기 때문이다.

3.2.2 단계 2-2 : 스페닝 트리의 개선



(그림 3) 스페닝 트리의 재구성

단계 2-2에서는 단계 2-1에서 생성한 초기 스페닝 트리를 개선한다. 우선 모든 노드 m 에 대해서 가까운 10개의 노드를 m_i ($1 \leq i \leq 10$)로 선정하고 $d(m)$ 과 m 의 후손 노드의 수 $h(m)$ 을 계산한다. 그리고 모든 노드 m 을 메모리 Q 에 $d(m)$ 의 역순으로 정렬한다.

단계 2-2의 핵심 동작은 지정 노드라고 부르는 특정한 노드 m 의 부모 노드를 선정하는 일이다. (그림 3.(a))는 현재의 스페닝 트리 T_O 이며 (그림 3.(b))는 재구성으로 고려되는 새로운 스페닝 트리 T_m 이다. $p_1(m)$ 과 m 의 후손 노드는 m_i 로 선정될 수 없다. 모든 후보 스페닝 트리 T_m 과 T_O 에 대해 식 (5)의 $L(T_m)$ 와 $L(T_O)$ 를 계산하고 가장 작은 L 을 생성하는 토폴로지를 선택한다. 만일 선택한 토폴로지가 T_m 이면 (그림 3.(b))처럼 트리를 재구성한다. 어느 토폴로지가 더 짧은지를 정할 때 우리는 토폴로지의 총길이를 구하지 않고 식 (7)에 정의된 토폴로지의 개선도 L_{mi} 를 사용한다. 이는 $d(\cdot)$ 가 단순한 거리 모델이기에 가능하다. 계산 결과 L_{mi} 가 양수면 새로운 토폴로지 T_m 가 이전 토폴로지 T_O 보다 우수함을 뜻한다. 만일 양수의 L_{mi} 가 여러 개 발생한다면 L_{mi} 가 가장 큰 토폴로지를 선택한다.

지정 노드 m 을 결정하고 나면 다음과 같이 재구성을 수행한다.

- 1) 가능한 모든 m_i 에 대해 식 (10)을 사용하여 L_{mi} 를 계산한다.
- 2) m_i 중에서 가장 큰 L_{mi} 를 생성하는 m_j 를 선택한다.
- 3) 만일 j 가 존재하지 않는다면 Q 에서 m 의 다음 노드를 지정 노드로 선택하고 1로 돌아간다.
- 4) (그림 3.(b))처럼 스페닝 트리를 재구성하고 다음 단계를 수행한다.
 - 4-1) $d(m)$ 을 dm 에 저장한다.
 - 4-2) 노드 m_j 와 m , m 의 모든 후손 노드에서 $d(\cdot)$ 와 $h(\cdot)$ 를 재계산한다. $d(\cdot)$ 의 역순으로 Q 를 재정렬한다.
 - 4-3) dm 보다 짧은 거리를 가진 노드 중에서 가장 큰 $d(m')$ 값을 가지는 노드 m' 를 다음 지정 노드로 선택하고 1로 돌아간다.

지정 노드 m 의 거리가 최대일 때부터 최소로 줄어들 때까지를 하나의 주기로 하자. 우리는 더 이상의 거리 개선이 없을 때까지 주기를 반복한다. 실제 시뮬레이션에서 두 번째 또는 세 번째 주기에서 계산이 종료되었다. 이는 FST가 비록 전역에서의 최적 해법은 아니지만 국지적으로 최적화된 해법을 안정적으로 제공하고 있음을 의미한다.

지금부터는 토폴로지의 개선도 L_{mi} 를 계산하는 방법을 소개한다. (그림 3)의 재구성이 일어나면 m 과 m 의 후손 노드의 거리만 변경된다. 거리는 토폴로지 상에서 결정되는데 재구성 시에는 신구의 두 가지 토폴로지인 T_O 와 T_N 이 언급되므로 혼란을 피하기 위해 아래의 식에서는 거리를 언급할 때 각 항 뒤에 사용되는 토폴로지를 명시하였다. 재구성이 일어나면 m 의 후손 노드 m' 의 거리 $d(m')$ 는

$((1-f) \cdot \delta(m)|_{T_N} - (1-f) \cdot \delta(m)|_{T_0})$ 의 차이가 생긴다. 만일 m 의 후손 노드 집합을 $Q(m)$ 이라 하면 식 (7)을 유도할 수 있다.

$$\begin{aligned} L_{mi} &= L(T_N) - L(T_0) \\ &= \sum_{\{q|q \in T_N\}} d(q)|_{T_N} - \sum_{\{q|q \in T_0\}} d(q)|_{T_0} \\ &= \sum_{\{q|q \in m \cup Q(m)\}} d(q)|_{T_N} - \sum_{\{q|q \in m \cup Q(m)\}} d(q)|_{T_0} \\ &= d(m)|_{T_N} + (1-f) \cdot h(m) \cdot \delta(m)|_{T_N} \\ &\quad - (d(m)|_{T_0} + (1-f) \cdot h(m) \cdot \delta(m)|_{T_0}). \end{aligned} \quad (7)$$

식 (7)에서 계산하기 복잡한 $d(\cdot)$ 항을 제거하기 위해서 식 (8)과 (9)를 정의하고 식 (7)의 $d(m)|_{T_N} - d(m)|_{T_0}$ 를 식 (8)과 (9)로 대체한 후 정리하면 식 (10)을 얻을 수 있다.

$$d(m)|_{T_N} = (1-f) \cdot \delta(m)|_{T_N} + f \cdot C(m, m_i), \quad (8)$$

$$d(m)|_{T_0} = (1-f) \cdot \delta(m)|_{T_0} + f \cdot C(m, p_1(m_i)). \quad (9)$$

$$\begin{aligned} L_{mi} &= (1-f) \cdot (1+h(m)) \cdot (\delta(m)|_{T_N} - \delta(m)|_{T_0}) \\ &\quad + f \cdot (C(m, p_1(m))|_{T_0} - C(m, m_i)) \end{aligned} \quad (10)$$

우리는 FST가 모든 f ($0 \leq f \leq 1$)에서 MST나 SPT보다 우수하길 바란다. 이를 위해 FST는 다음 두 정리를 만족해야 한다. 두 정리의 증명은 매우 유사하므로 정리 1만을 증명하였다.

정리 1. f 가 1일 때에 FST는 최적치인 MST와 동일하다.

(증명) 단계 1에서 FST 알고리즘은 전염전일 때 초기 트리로 MST를 생성한다. 초기 트리가 이미 최적이기 때문에 우리는 어떤 지정 노드 m 에 대해서도 양수의 L_{mi} 를 찾을 수 없으므로 단계 2-2에서 재구성은 발생하지 않는다. 그 결과 FST는 MST로 종료된다. ■

정리 2. f 가 0일 때에 FST는 최적치인 SPT와 동일하다.

4. 시뮬레이션 및 분석

우리는 시뮬레이션을 통해 다양한 f 에서 식 (4)와 (5)에 따라 FST와 SPT, MST의 토폴로지 길이를 비교한다. 우리

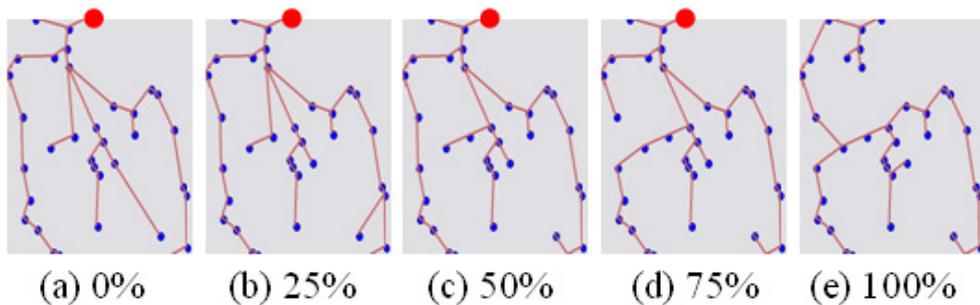
는 동일한 크기의 정방형 센서 필드에 25-노드와 100-노드, 400-노드를 임의로 분포시키고 루트 노드를 임의로 선정하였다. 결과값은 각각 100개의 WSN에 대한 평균값을 사용하였다. WSN에서의 링크 비용 $C(i,j)$ 는 거리의 제곱을 사용하였고 거리의 네제곱의 경우도 별도로 고려하였다. 이는 무선 통신에서 전송 거리 l 이 주어질 때에 전송 손실 모델이 $l^2 \sim l^4$ 에 비례하기 때문이다.

(그림 4)는 35-노드 WSN에서 f 가 0에서 1까지 변화할 때 FST의 형태를 보여준다. 루트 노드 R 은 WSN의 상단 중간에 큰 점으로 표시하였다. (그림 4.(a))와 (그림 4.(e))는 각각 SPT와 MST를 나타낸다. SPT의 링크들은 R 로 향하는 직선 경로를 선호하기 때문에 (그림 4.(a))와 같은 토폴로지를 형성하고 (그림 4.(e))의 MST의 링크와 비교해보면 길이가 다소 긴 특징을 보인다. f 가 0에서 0.75까지 변화할 때 토폴로지의 형태는 크게 바뀌지 않지만 0.75에서 1로 증가할 때 많은 변화가 발생함을 관찰할 수 있다.

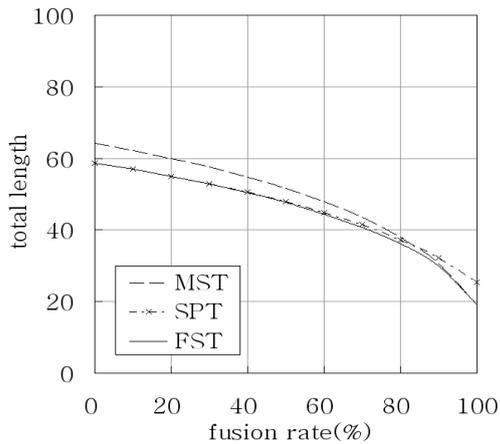
(그림 5)에서 (그림 8)은 f 의 전체 범위($0 \leq f \leq 1$)에 대해 식 (5)에서 정의한 토폴로지의 총길이를 나타낸다. (그림 5)에서 (그림 7)은 노드의 수가 각각 25와 100, 400일 때 SPT와 MST, FST의 토폴로지 총길이를 보여준다. 우리는 모든 경우에서 f 가 커질수록 식 (4)에서 정의한 거리가 짧아지기 때문에 토폴로지의 총길이가 또한 줄어들음을 알 수 있다. 또한 FST는 모든 f 에서 SPT와 MST보다 총길이가 짧고 f 가 약 0.9일 때 SPT와 MST 곡선이 교차함을 확인하였다. 두 곡선의 차이는 노드 수가 증가하거나 노드의 밀집도가 높아질수록 커진다.

(그림 8)에서 우리는 추가적으로 $C(i,j)$ 에 거리의 네제곱을 할당하여 조사하였다. (그림 8)에서 토폴로지 총길이는 f 가 증가함에 따라 크게 줄어들어서 y 축을 \log 단위로 사용했다. \log 단위를 제외하면 (그림 5)에서 (그림 8)은 모두 유사한 곡선 형태를 갖는다.

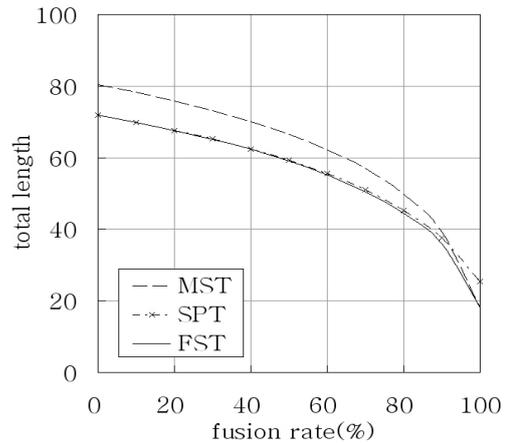
<표 1>은 f 가 20% 단위로 변화할 때 FST에 대한 SPT와 MST의 상대적인 총길이를 비교한다. f 가 93%일 때의 값은 SPT와 MST 곡선이 교차하는 시점이기 때문에 특별히 추가하였다. <표 1>에서 마지막의 평균값을 제외한 모든 열은 FST에 대한 토폴로지의 상대적인 총길이를 나타낸다. 마지막 행의 MSPT는 MST와 SPT 중에서 토폴로지의 총



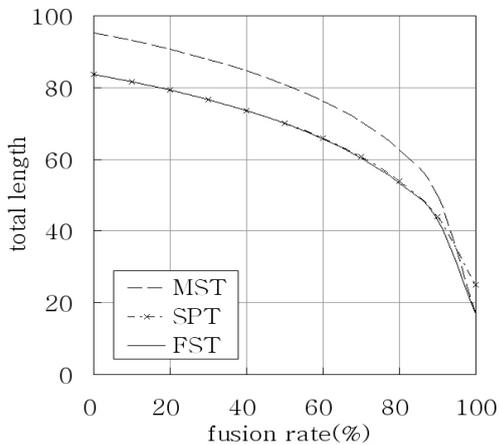
(그림 4) 다양한 f 값에 따른 FST 토폴로지의 변화



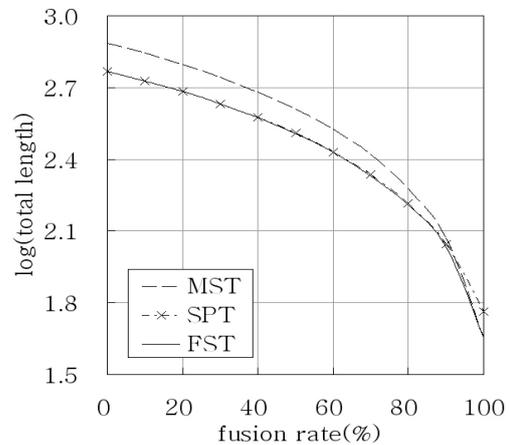
(그림 5) 25-노드 WSN에서 f 의 변화에 따른 MST와 SPT, FST의 토폴로지 총길이



(그림 6) 100-노드 WSN에서 f 의 변화에 따른 MST와 SPT, FST의 토폴로지 총길이



(그림 7) 400-노드 WSN에서 f 의 변화에 따른 MST와 SPT, FST의 토폴로지 총길이



(그림 8) 100-노드 WSN에서 f 의 변화에 따른 MST와 SPT, FST의 토폴로지 총길이 ($\alpha(i,j)$ 를 거리⁴로 재정의)

<표 1> 100-노드 WSN에서 FST와 SPT, MST의 상대적인 총길이 비교

토폴로지 \ $f(\%)$	0	20	40	60	80	93	100	평균
FST	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SPT	100.0	100.1	100.5	101.5	104.2	110.3	140.7	108.2
MST	148.0	146.0	143.9	138.4	128.5	111.4	100.0	130.9
MSPT*	100.0	100.1	100.5	101.5	104.2	110.3	100.0	102.4

* MSPT는 SPT와 MST 중 더 짧은 총길이를 가지는 토폴로지이다.

길이가 더 짧은 토폴로지이다. <표 1>의 평균값을 보면 FST의 토폴로지 총길이는 약 MST보다 31%, SPT보다 8% 상대적으로 짧다. 또한 FST는 SPT와 MST 중 우수한 방식을 택한 경우(MSPT)보다도 2.4% 우수하다.

5. 결론

SPT와 MST는 각각 무퓨전일 경우와 전퓨전일 경우 에

너지를 최소로 소비하기 때문에 제한된 에너지를 사용하는 센서 네트워크에서 매우 유용한 토폴로지이다. 우리는 토폴로지의 총길이를 측정할 때 퓨전율 f 를 고려하는 새로운 방법과 one-time 퓨전 모델을 제안하였다. 또한 토폴로지의 총길이를 재정의하고 f 를 확장하여 MST와 SPT를 모두 수용할 수 있는 새로운 토폴로지 FST를 제안하였다.

FST는 많은 장점을 가지고 있다. 첫째, FST는 통계에 의해 쉽게 f 를 계산할 수 있다. 둘째, FST는 우리가 고려한

모든 WSN 환경에서 MST와 SPT보다 우수하였고 무퓨전과 전퓨전일 때에도 최적이었다. 시뮬레이션 결과 100-노드 WSN에서 FST의 총 길이는 전체 f 에 대한 평균값에서 약 MST보다 31%, SPT보다 8% 짧았다. 따라서 우리는 FST가 convergecast를 사용하는 센서 네트워크에서 매우 유용함을 확인하였다.

참 고 문 헌

[1] Jamal N. Al-Karaki and Ahmed E. Kamal, "Routing Techniques in Wireless Sensor Networks : A Survey," IEEE Wireless Communications, Dec., 2004.
 [2] Lan F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, Aug., 2002.
 [3] J. N. Al-Karaki et al., "Data Aggregation in Wireless Sensor Networks - Exact and Approximate Algorithms," Proc. IEEE Wksp. High Perf. Switching and Routing 2004. Phoenix, AZ, April, 2004.
 [4] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor networks: a Survey," IEEE Communication Magazine, Vol.44, No.4, April, 2006.
 [5] L. Klein. "Sensor and Data Fusion Concept and Applications," Artech House, Boston, Mar., 1993.
 [6] S. Upadhyayula and S. K. S. Gupta, "Spanning Tree based Algorithms for Low Latency and Energy Efficient Data Aggregation Enhanced Convergecast (DAC) in Wireless Sensor Networks," Ad Hoc Networks, Vol.5, No.5, 2007.
 [7] H. Luo, Y. Liu, and S.K. Das, "Adaptive Data Fusion for Energy Efficient Routing in Wireless Sensor Networks," IEEE Trans. Mobile Computing, Oct., 2006.
 [8] H. Luo, J. Luo, Y. Liu, and S.K. Das, "Routing Correlated Data with Fusion Cost in Wireless Sensor Networks," IEEE Trans. Mobile Computing, Nov., 2006.
 [9] H. Cormen, E. Leiserson, L. Rivest, and Stein, "Introduction to Algorithms- second edition," MIT press, 2001.

부 록

단계 1 : 최적 퓨전율의 계산

이 장에서는 주어진 스페닝 트리 T_S 를 사용하여 최적의 퓨전율 f_o 를 계산하는 방법을 소개한다. 우리는 스페닝 트리의 모든 링크에서 실제 전송되는 데이터와 one-time 퓨전 모델의 전송 데이터의 총길이를 계산하여 두 값의 차이를 최소화하는 f_o 를 찾는다. One-time 퓨전 모델을 사용할 경우에 두 길이의 차이가 f 에 대한 이차식이기 때문에 차이를 최소화하는 최적의 f 값인 f_o 를 항상 찾을 수 있다.

편의를 위하여 모든 노드가 동일한 길이의 데이터를 생성한 후에 T_S 를 따라 전송된다고 가정하자. 그리고 E_{ij} 를 T_S 에

서 임의의 링크 $l(i,j)$ 로 전송되는 데이터의 총길이이라 하자. One-time 퓨전 모델을 적용하여 동일한 문제를 풀면 (그림 1)과 같이 $l(i,j)$ 로 전송되는 데이터의 가상 길이 W_{ij} 를 계산할 수 있다. One-time 모델은 간단하기 때문에 W_{ij} 는 식 (11)과 같이 f 에 대한 일차식으로 표현할 수 있다.

$$W_{ij} = a_{ij} \cdot f + b_{ij}, \quad a_{ij}, b_{ij} \text{는 상수.} \quad (11)$$

$l(i,j)$ 에서 one-time 퓨전 모델의 오차값은 $|E_{ij} - W_{ij}|$ 이다. f_o 는 스페닝 트리 T_S 에 포함되는 모든 링크 중에서 $|E_{ij} - W_{ij}|^2$ 를 최소화하는 최적의 f 이다. 식 (12)의 우항은 f 를 제외하고 모두 상수값이므로 상수 α, β, γ 의 세 값으로 이차식을 정리할 수 있다.

$$\begin{aligned} \sum_{\{(i,j)|l(i,j) \in T_S\}} (E_{ij} - W_{ij})^2 &= \sum_{\{(i,j)|l(i,j) \in T_S\}} \{a_{ij} \cdot f + (b_{ij} - E_{ij})\}^2 \quad (12) \\ &= \alpha \cdot f^2 - \beta \cdot f + \gamma, \quad \alpha, \beta, \gamma \text{는 상수.} \end{aligned}$$

α, β, γ 가 주어진 경우에 f ($0 \leq f \leq 1$)의 범위에서 최소값 f_o 를 계산할 수 있다.

$$\begin{aligned} \min \sum_{\{(i,j)|l(i,j) \in T_S\}} (E_{ij} - W_{ij})^2 &= \min(\alpha \cdot f^2 - \beta \cdot f + \gamma) \\ &= \min(\alpha \cdot (f - \frac{\beta}{2\alpha})^2 - \frac{\beta^2}{4\alpha} + \gamma) \quad (13) \\ &\geq \alpha \cdot f_o^2 - \beta \cdot f_o + \gamma. \end{aligned}$$

식 (13)의 퓨전을 오차는 우항의 제곱항이 0일 때가 최소이므로 최적의 퓨전율 f_o 는 다음과 같다.

$$f_o = \frac{\beta}{2\alpha}. \quad (14)$$



서 창 진

e-mail : cjsuh@ssu.ac.kr
 1982년 서울대학교 제어계측학과(학사)
 1984년 서울대학교 대학원 제어계측학과(공학석사)
 1985년~1990년 한국전자통신연구원 선임연구원

1996년 ECE(Electric and Computer Engineering), Univ. of Massachusetts at Amherst(공학박사)
 1997년~현 재 숭실대학교 컴퓨터학부 교수
 관심분야 : Sensor network, Carrier Ethernet, Switching, Routing theory, BGP 등



신 지 수

e-mail : jsshin@networking.com

2003년 숭실대학교 컴퓨터학부(학사)

2005년 숭실대학교 대학원 컴퓨터학과
(공학석사)

2005년~현 재 숭실대학교 컴퓨터학과
박사과정

관심분야: Sensor network, Carrier Ethernet, Routing theory,
Ubiquitous network 등