

유전자 알고리즘 기반의 센서 네트워크 토폴로지 제어

박충명[○], 김동국, 이좌형, 김 윤, 정인범
강원대학교 컴퓨터정보통신공학과

{ cmpark[○], dkkim, jhlee, yooni, ibjung }@snslab.kangwon.ac.kr

Topology Control based on Genetic Algorithm in Sensor Network

Chong-Myung Park[○], Dong-Gug Gim, Joa-Hyoung Lee, Yoon Kim, In-Bum Jung
Dept. of Computer Information & Telecommunication Engineering, Kangwon National Univ.

요 약

많은 센서 노드가 환경정보를 수집하는 이벤트 기반의 센서 네트워크는 국부적인 이벤트 발생에 따라 혼잡이 발생할 수 있다. 네트워크의 혼잡으로 인해 중요한 정보의 손실이 일어날 수 있으며, 과도한 RF 모듈의 사용으로 네트워크의 수명이 단축될 수 있다. 본 논문에서는 이벤트가 발생하여 네트워크 트래픽이 증가할 때, 트래픽이 집중된 노드에서 주변 노드들의 데이터 전송률을 고려하여 자식 노드들의 트래픽을 분산 시키는 유전자 알고리즘 기반의 트래픽 분산 기법을 제안한다.

1. 서 론

센서 네트워크에 대한 관심이 증가하면서 다양한 시나리오를 바탕으로 어플리케이션이 개발되고 있다. 예를 들면, 산불 감시, 동식물의 생태 감시와 같은 환경 모니터링 어플리케이션과, 사람이나 기계 또는 건축물의 움직임을 감지하는 움직임 모니터링 어플리케이션 등이 있다. 이러한 대부분의 센서 네트워크는 노드들이 배치된 후 관리자가 노드의 유지관리를 하지 않는 형태이므로 한번 배치된 후 긴 수명을 갖고 데이터를 수집하기 위해서는 노드 자체의 전력을 효율적으로 사용하여야한다[1].

네트워크의 수명 연장을 위해 각 노드는 수집한 데이터의 전송 횟수를 줄여야할 필요가 있다[2]. 이를 위해 어플리케이션에서 특정 이벤트가 감지되지 않는 경우엔 긴 주기로 데이터를 수집하여 전송하게 하고, 이벤트를 감지할 경우 짧은 주기로 데이터를 수집하여 이벤트에 대한 분석을 할 수 있는 충분한 양의 데이터를 전송하는 방법을 사용할 수 있다. 예를 들면, 교량 안전 관리를 위한 센서 네트워크에서 정상 상태의 경우 초당 1패킷을 전송하게 하고 이상 상태를 감지하였을 경우에는 초당 100패킷을 전송하게 하여 정밀한 분석을 가능하게 할 수 있다. 일부 지역에서 생성되는 데이터의 증가는 네트워크의 혼잡 상태를 야기시킬 수 있다[3].

네트워크의 혼잡은 과도한 RF 모듈의 사용으로 네트워크의 수명을 단축시킬 수 있을 뿐만 아니라 멀티홉을 이용하여 싱크로 데이터를 전송 중 중요한 정보의 손실을 발생시킬 수 있으므로 이에 대한 해결 방안이 필요하다[4].

본 논문에서는 센서 네트워크에서 특정 노드에 트래픽이 집중되는 경우, 집중된 트래픽을 주변 노드들로 분산시키는 알고리즘인 Topology Control based on Genetic Algorithm(TCGA)을 제안한다. TCGA는 트래픽이 집중된 노드의 자식 노드들로부터 주변 노드들의 정보를 수집하여 자식 노드들의 데이터 전송을 특정 비율로 다른 노드로 포워딩 시킨다. 센서 노드들이 밀집되어 주변 노드들이 많고 각 노드들의 데이터 전송률이 다양할 경우 자식 노드들에 적당한 포워딩 노드와 전송률을 선택하기 어렵다. TCGA는 적절한 포워딩 노드와 전송률을 선택하여 주변 노드들에 트래픽을 공평하게 분배할 수 있도록 유전 알고리즘을 사용한다.

유전자 알고리즘은 두 부모 유전자로부터 자손 유전자를 생성하고 변이시켜 더 나은 형질을 가진 유전자를 보존 및 진화 시키는 알고리즘이다[5]. 유전자 알고리즘은 초기 집단을 구성하고 유전 연산자와 적합도(fitness) 함수를 이용하여 적합도가 우수한 개체들을 선택하여 점진적으로 개선된 집단을 형성한다. 본 연구에서는 자식 노드와 그 주변노드들의 데이터 전송률을 이용하여 유전자를 정의하고 주변 노드들의 데이터 전송률이 공평성을 유지할 수 있도록 적합도 함수를 설계하여 이를 평가한다.

본 논문의 구성은 다음과 같다. 2장에서는 유전자 알고리즘에 대해 살펴보고, 3장에서는 제안하는 알고리즘인 TCGA를 설명한다. 4장에서는 실험을 통해 TCGA의 성능을 측정하고, 5장에서는 실험의 결과를 분석하고 본 논문의 결론을 맺는다.

2. 유전자 알고리즘 (Genetic Algorithm : GA)

유전자 알고리즘은 복잡한 문제 영역을 확률적인 방법으로 탐색하여 해를 도출해내는 방법으로 기계 학습, 로봇 공학, 순회 판매원 문제를 비롯한 최적화 문제와 다양한 분야에서 사용되고 있다[6].

그림 1은 기본적인 단순 유전 알고리즘의 흐름을 나타낸다. 우선 복잡한 문제의 해를 표현한 염색체들의 집합을 만들고 세대를 거듭하여 교차와 변이의 유전 연산을 수행한 후 각 개체의 적합도를 평가하여 우수한 개체들로 다음 세대를 구성한다.

유전자 알고리즘을 구현하기 위해서는 우선 문제에 맞는 염색체 표현 방법과 교차, 변이와 같은 유전 연산자의 설계가 우선되어야 한다. 또한 문제에 맞는 적합도 함수를 설계하는 것이 가장 중요하다. 염색체를 표현하는 방법에는 이진수 표현, 순열 표현, 실수 표현 등이 있으며, 유전 연산자에는 선택, 교차, 변이, 대치 연산자 등이 있다. 본 논문에서는 센서 노드의 특성을 고려하여 센서 노드들의 노드 번호와 데이터 전송률, 포워딩 전송률을 이용하여 유전자를 표현하였으며, 각 개체의 포워딩 전송률을 새로 할당하거나 이웃 노드의 정보를 교환함으로써 유전연산을 수행하였다.

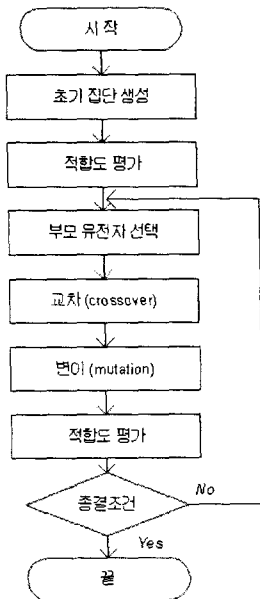


그림 1. 단순 유전자 알고리즘.

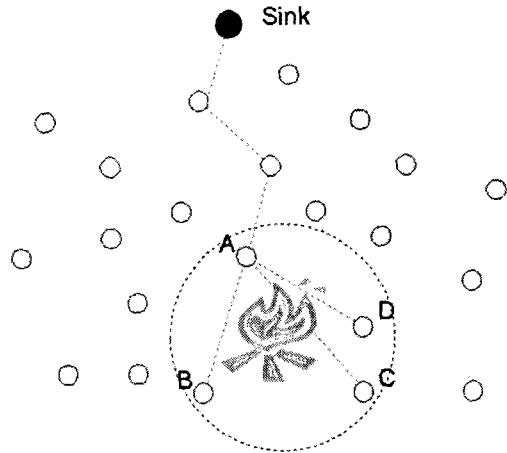


그림 2. 지역적인 이벤트 감지.

3. 유전자 알고리즘을 이용한 토폴로지 제어

센서 네트워크에서 특정 노드에 트래픽이 집중되는 경우, 집중된 트래픽을 주변 노드들로 분산시킴으로써 혼잡한 네트워크로 인해 발생할 수 있는 데이터 손실 문제를 해결할 수 있다. 또한 데이터의 전송량이 한 노드에 집중되지 않고 여러 노드에서 분담하여 전송함으로써 네트워크의 수명을 증가시킬 수 있다. 그림 2는 앞서 설명한 바와 같이 긴 샘플링 주기를 갖는 센서 네트워크에서 이벤트 발생 시 일부 지역에 데이터 전송량이 증가함을 나타낸다. 그림 2에서 화재 발생시 A~D 노드가 이를 감지하고 라우팅 경로가 정선과 같다면, A~D 노드는 화재를 감지했으므로 샘플링 주기를 증가시킨다. 이때 증가된 데이터는 A 노드에 집중된다. 이러한 경우 A 노드에서 자식 노드인 B~D 노드의 데이터 전송을 가용한 이웃 노드들로 분산시킬 필요가 있다.

3.1 TCGA의 동작 방법

TCGA 알고리즘이 동작하기 위해서는 주변 노드들의 전송률에 관한 정보를 저장하고 있어야 한다. 이를 위해 각 노드들은 주기적으로 자신의 정보를 브로드캐스트하고 이웃 노드들은 수신한 정보를 저장한다.

그림 3은 TCGA 알고리즘의 시퀀스 다이어그램이다. 다이어그램에서 BN은 그림 2의 A노드에 해당한다. A노드의 데이터 전송률이 증가하게 되면 A노드는 자식 노드에 자신의 상태를 알리는 메시지(busy msg)를 전송한다. 자식노드들은 저장하고 있던 이웃노드들의 정보를 부모 노드에게 전송한다. 부모노드는 자식노드들로부터 수신

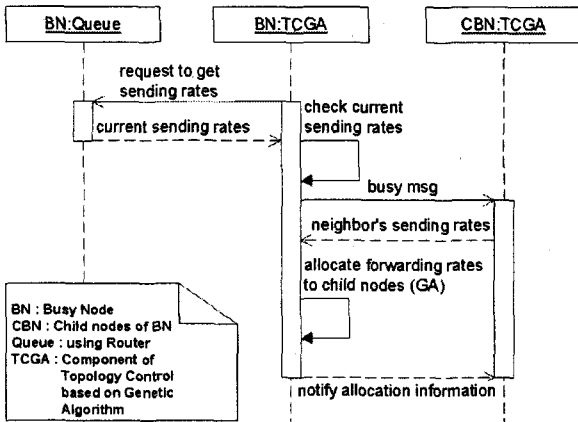


그림 3. TCGA 시퀀스 다이어그램.

한 정보를 이용하여 유전자 알고리즘을 수행한 후 적합도가 높은 개체를 선택하여 개체가 포함하고 있는 포워딩 정보를 자식 노드들에게 전송한다. 자식 노드들은 포워딩 정보를 확인하고 할당된 비율로 부모노드 대신 포워딩 노드에 데이터를 전송한다.

그림 4는 BN에서 유전자 알고리즘을 이용하여 자식 노드들에 포워딩 노드와 전송률을 할당하는 함수를 나타낸다. 초기에 자식노드들로부터 수신한 이웃노드 정보를 이용하여 유전자 알고리즘을 이용한 해집합을 생성한다. 그리고 종결조건을 만족할 때까지, 해집합에서 두 개체를 선택하여 교차, 변이, 적합도 평가, 대치 함수 수행을 반복한다. 마지막으로 해집합에서 적합도가 높은 개체를 선택하여 자식 노드로 전송한다. 위와 같은 유전자 함수들을 수행함으로써 세대가 지날수록 해집합에는 적합도가 낮은 개체들은 대치되고 높은 개체들만 남게 된다.

```

allocFWRateGA ( ) {
  initialize()
  FOR i=0 TO end condition DO
    select 2 chromosomes
    crossover() // generate new chromosome
                with 2 chromosomes
    mutation() // allocate new forwarding rate
    evaluate() // evaluate new chromosome
    substitution() // replace chromosomes
  ENDFOR
  SEND chromosome with the best fitness TO child nodes
}
    
```

그림 4. 유전자 알고리즘을 이용한 포워딩 정보 할당.

3.2 염색체 표현 방법

유전자 알고리즘을 구현하기 위해서는 주어진 문제를 해결하기 위해 개체로 사용되는 염색체를 표현하는 방법이 필요하다. 염색체 표현 방법에는 이진수 표현, 순열 표현, 실수 표현 등이 있다. 본 연구에서는 특정 노드에 집중되는 트래픽을 분산하기 위해 다음과 같은 실수 표현 방법으로 염색체를 표현하였다.

FIT	CID[n]	CR[n]	FR[n]	NID[2][n]	NR[2][n]
-----	--------	-------	-------	-----------	----------

- FIT : 적합도
- CID : 자식노드 번호
- CR : 자식노드의 데이터 전송률
- FR : 포워딩 전송률
- NID : 이웃노드 번호
- NR : 이웃노드의 데이터 전송률

3.3 적합도 함수

본 연구에서 제안하는 알고리즘은 네트워크 토폴로지를 변경하여 특정 노드에 집중된 트래픽을 분산함으로써 네트워크 혼잡으로 인한 데이터 손실을 방지한다. 따라서 유전자 알고리즘의 적합도 함수는 이러한 목적을 달성할 수 있도록 구현되어야 하며, 다음의 식(2)와 같이 설계 되었다.

$$ANR = \sum_{i=0}^{n-1} (NR_i + FR) / n \quad (1)$$

$$Fitness^{-1} = \sqrt{\sum_{i=0}^{n-1} \{ANR - (NR_i + FR)\}^2 / n} \times ANR \quad (2)$$

식(1)의 ANR은 해집합의 한 개체에 포함된 이웃노드들의 데이터 전송률과 새로 할당된 포워딩 전송률의 합의 평균을 나타내며, 식(2)에서 적합도는 ANR을 이용하여 계산한 표준편차와 ANR의 곱의 역수이다. 적합도는 각 노드들의 전송률의 평균과 표준편차에 반비례한다. 표준편차가 크다는 것은 데이터 전송률이 고르게 분산되지 않았음을 의미하며, ANR이 큰 것은 전체적으로 전송해야할 평균 데이터양이 많음을 나타낸다. 표준편차와 ANR을 같이 적합도 함수에 사용한 것은 표준편차만을 적합도 함수로 사용할 경우, 노드들에 할당된 전송률은 작지만 표준편차가 큰 개체와 할당된 전송률은 많지만 표준편차가 작은 개체가 있을 경우 후자가 선택될 확률이 높기 때문이다.

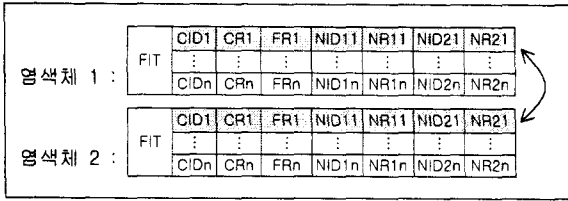


그림 5. 교차 연산.

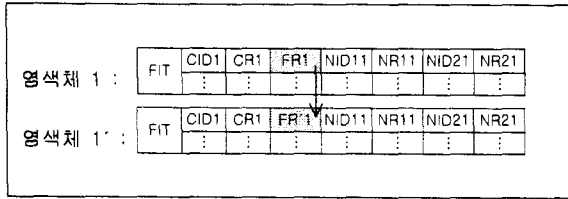


그림 6. 변이 연산.

3.4 유전 연산

교차, 변이 연산의 유전 연산을 반복적으로 수행함으로써 해집합 내의 염색체들은 점진적으로 적합도가 높아진다. 그림 5, 6은 제안 기법에 사용된 교차, 변이 연산을 나타낸다.

교차 연산자는 그림 5와 같이 해집합에서 두 부모 염색체를 선택하고 염색체의 각 원소 중 임의의 자식 노드에 관한 정보를 교환하여 새로운 염색체를 생성한다. 염색체는 자식노드(CID, CR)와 자식노드의 이웃노드(NID, NR)들에 관한 정보를 쌍으로 가지고 있는데, 이 정보를 다른 염색체와 교배하는 것을 의미한다.

변이 연산자는 그림 6과 같이 염색체의 포워드 전송률(FR)을 재할당하여 개체의 특성을 변이시킨다. 염색체는 자식노드와 그 이웃노드들의 정보를 가지고 있고 유전자 알고리즘에서 할당된 FR을 포함하고 있다. 이 FR을 어떻게 할당하는가에 따라서 염색체에 포함되어 있는 노드들의 전송률이 결정되며 변이에 의해 적합도가 변하게 된다.

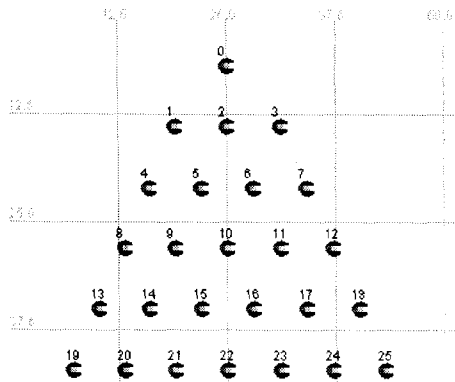


그림 7. 시뮬레이션 토폴로지.

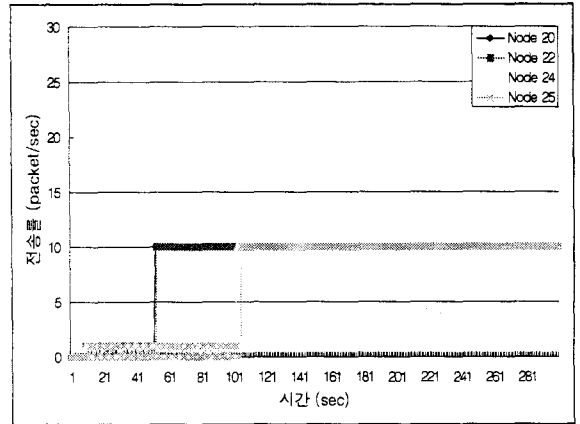


그림 8. 하위 노드의 전송률 변화.

4. 실험 및 성능 평가

제안 알고리즘의 성능 평가를 위해 TinyOS와 TOSSIM 시뮬레이터를 사용한 모의 실험을 수행하였다 [7]. 포토센서로 데이터를 샘플링하여 싱크노드(0번 노드)로 전송하는 Surge 어플리케이션과 멀티홉 라우팅을 사용하였다. 그리고 제안 알고리즘을 구현한 TCGA 컴포넌트를 Surge 어플리케이션에 연결하고, 26개의 센서 노드들을 그림 7과 같이 배치하여 실험하였다[7].

데이터 전송률은 최초 초당 0.5개의 패킷을 전송하도록 하고, 그림 8과 같이 시뮬레이션 시작 후 약 50초에 20, 22번 노드, 100초에서 24, 25번 노드의 데이터 전송률이 증가하도록 하였다.

그림 9는 TCGA를 적용하지 않은 센서 네트워크에서 상위 노드인 1~3번 노드의 데이터 전송률 변화를 나타낸다. 라우팅 컴포넌트에서 노드들의 경로를 설정하는데 2번 노드는 자식노드를 가지고 있지 않아 데이터 전송률이 거의 없다. 반면에 1, 3번 노드들은 자식노드들로부터 데이터를 전송받아 싱크로 전송하기 때문에 하위 노드들의 전송률이 증가하는 시점에서 1번 노드는 26 packet/s, 3번 노드는 23 packet/s의 전송률을 나타내어 전송률이 불균형적으로 고정되는 것을 알 수 있다.

TCGA를 적용한 센서 네트워크의 경우 그림 10과 같은 변화를 보였다. 50초에 20, 22번 노드의 전송률 증가에 따라 1번 노드의 전송률이 급격하게 증가하고 1번 노드에서 TCGA 알고리즘이 동작하게 된다. 따라서 1번 노드의 전송률은 감소하며 이웃노드인 2번 노드의 데이터 전송률이 증가하게 된다. 또한 100초에 24, 25번 노드의 전송률이 증가하여 상위 노드인 3번 노드의 전송률이 급격하게 증가한다. 1번 노드와 마찬가지로 3번 노드에서

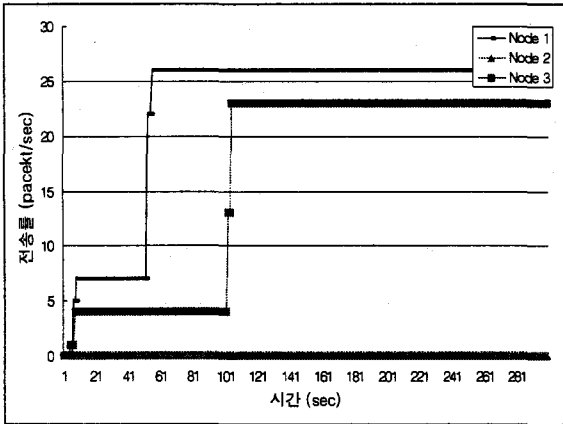


그림 9. 시간에 따른 전송률 변화 (w/o TCGA).

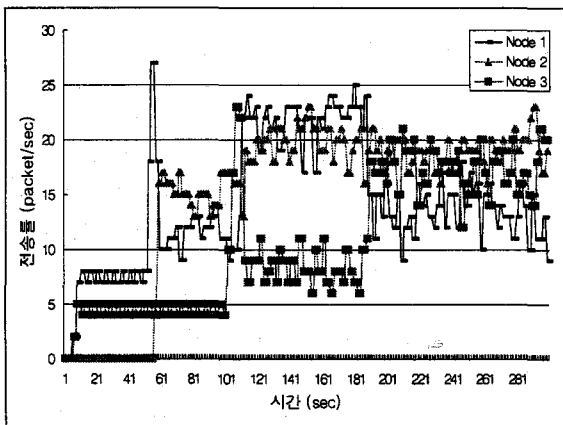


그림 10. 시간에 따른 전송률 변화 (w/ TCGA).

TCGA 알고리즘이 동작하며 이때 1번 노드와 2번 노드로 트래픽이 분산되어 3번 노드의 데이터 전송률이 감소하는 것을 나타내었다. 200초 이후로는 TCGA 컴포넌트가 유전자 연산을 포함한 일련의 동작을 수행하여 각 노드의 전송률 분배가 이루어짐을 확인할 수 있다.

5. 결론 및 향후 계획

본 논문에서는 센서 네트워크에서 지역적인 이벤트 감지 등에 의해 특정 노드에 트래픽이 집중되는 경우, 집중된 트래픽을 주변 노드들로 분산시키는 TCGA 알고리즘을 제안하고 실험을 통해 성능을 확인하였다.

실험 결과에서 같은 레벨에 있는 센서 노드들의 전송률이 변화하면서 평균적으로 비슷한 수준의 전송률을 유지하는 것을 보였으며 이는 집중된 트래픽이 주변 노드로 분산됨을 나타낸다.

현재 TCGA는 주변 노드들과 자신의 전송률만을 고려하여 알고리즘을 수행한다. 향후에는 전송률과 함께 노드들의 전력 상태를 고려할 예정이며, 효율적인 트래픽의 분산을 위해 센서 네트워크에 적합한 유전자 알고리즘의 확률 모델에 대한 연구를 진행할 계획이다.

6. 참고 문헌

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, p393-422, 2002.
- [2] J. Polastre, R. Szewczyk, C. Sharp, D. Culler, "The Mote Revolution: Low Power Wireless Sensor Network Devices," in *Proceedings of Hot Chips 16: A Symposium on High Performance Chips*, August 2004.
- [3] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," In *proceedings of the ACM MobiHoc Conference*, 2003.
- [4] J. Kang, Y. Zhang, B. Nath, S. Yu, "Adaptive Resource Control Scheme to Alleviate Congestion in Sensor Networks," In *Proceedings of BASENETS*, October 2004.
- [5] D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley Publishing Company, Inc. 1989.
- [6] <http://cs.felk.cvut.cz/~xobitko/ga/>
- [7] <http://tinyos.net>