

Ad Hoc 네트워크에서 효율적 트리 생성 알고리즘

조영진, 송준현, 박소담, 박정흠, 박승진
 명지대학교 컴퓨터 공학과

e-mail : parksinfo@hanmail.net, cosmos_901@naver.com

An Efficient Tree Generation Algorithm in Ad Hoc Networks

Young-Jin Cho, Jun-Hyeon Song, So-Dam Park, Jung-Heum Park, Seung-Jin Park
 Dept. of Computer Engineering, Myung-Ji University

요 약

Ad-hoc 네트워크상에서 여러 라우팅 기법들이 제안 되었으며, 그 중 근래에 발표된 트리 토폴로지에 바탕을 둔 AIR(Automatic Incremental Routing)는 그 단순성과 효율성에서 뛰어난 알고리즘으로 주목 받고 있다. 하지만 AIR 와 같이 트리에 바탕을 둔 알고리즘은 형성된 트리의 형태에 따라 그 효율성이 크게 좌우 되는 반면, 트리 형성방법은 많이 논의 되지 않았다. 본 논문은 AIR 상에서의 라우팅 방법을 이용하여 효율적인 트리의 형성 알고리즘을 제안하고, 이 알고리즘으로 생성된 트리, 임의로 생성된 트리, 그리고 최적의 트리를 각각의 Wiener 수를 비교하여 그 효율을 입증한다.

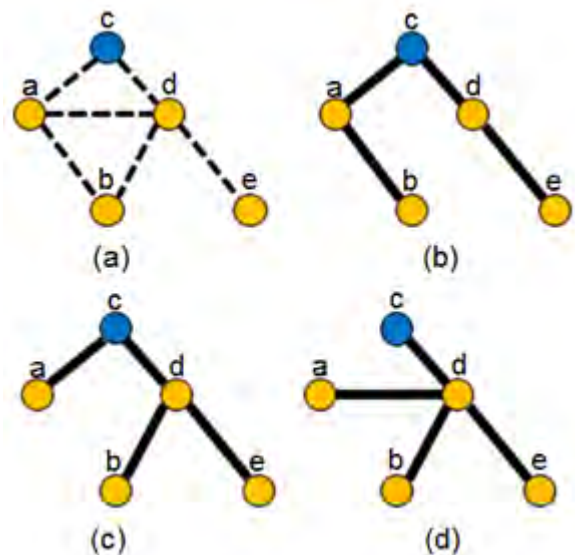
1. 서론

현재 무선네트워크는 Access Point(AP) 를 이용한 정적인 무선네트워킹 방식에서 AP 없이 노드간 자율적으로 통신하며 네트워크 토폴로지가 동적으로 변하는 ad-hoc 네트워크(MANET)같은 방식으로 개선시키고자 하는 노력이 지속되고 있으며, 따라서 이에 맞는 라우팅 방법들이 많이 연구되고 있다. 그 중 하나로 소개된 AIR[1, 2, 5]는 하나의 루트노드로 부터 시작되어 트리 형태의 네트워크 토폴로지를 생성하며 노드들에게 각각 prefix label(PL)을 부여, 이 PL 정보를 통해 통신을 하는 방법이다. 이는 경우에 따라 다양한 네트워크 토폴로지가 나올 수 있고, 토폴로지마다 네트워크의 효율도 천차만별이지만, 이전 AIR 에 관한 논문에서는 어떤 방식으로 트리 토폴로지를 구성하느냐에 대한 논의가 분명하지가 않다. 이 논문에서 우리는 이 효율성에 대한 판단 기준을 정하고 또 이 기준에 합당한 새로운 알고리즘을 제시하며, 이 알고리즘을 통해 생성된 네트워크 토폴로지의 효율성을 객관적으로 판단하여 그 우수성을 보여준다

본 논문의 구성은 다음과 같다. 2 장에서는 이 논문에서 사용되는 트리 토폴로지의 효율성 측정기준을 설명하고, 3 장에서는 새로운 알고리즘에 대해서 설명하며, 4 장에서는 이 알고리즘을 통해 생성된 네트워크 토폴로지의 효율성을 Wiener 수[3, 4]를 바탕으로 비교 분석한 뒤 5 장에서는 결론과 앞으로 제안된 알고리즘이 개선해야 할 부분에 대해서 논의 한다.

2. 트리구조의 기본 설명과 효율성 비교 기준

그림 1 (a)에서 점선들은 서로의 전파가 도달하는 노드들은 표시하고 있고, 이에 따라 그림 1 (b), (c), (d) 와 같은 여러 트리 구조가 형성될 수 있다. 노드의 degree 는 그 노드에 연결된 노드의 수를 말하며 예를 들어 그림 1 (c)에서 노드 a, b, c, d, e 의 degree 는 각각 1, 1, 2, 3, 1 이다.



(그림 1) (a)는 노드들과 전송반경 표시 점선, (b) (c) (d) 는 (a)로 부터 형성될 수 있는 여러 형태의 트리구조.

AIR 에서 각 노드들은 자신의 고유 ID 이외에 다음과 같은 재귀적 방식으로 PL 을 가진다. 첫째, root

노드를 선택하고 PL=0 를 부여한다. PL 을 부여받은 노드는 자식노드들 중 아직 PL 을 부여받지 못한 노드들에게 자신의 PL 에 {1, 2, 3, ... }중 하나를 첨부하여 유일한 PL 을 부여한다. 예를 들어, 그림 1 의 (c) 에서 c 를 root 노드라 하면, c 의 PL 은 0 이고, a 와 d 는 {01, 02}중 각각 다른 하나를, b 와 e 는 {021, 022} 중 각각 다른 PL 을 갖는다.

AIR 에서와 같이 라우팅이 트리를 바탕으로 할 때 효율적인 트리 생성은 네트워크의 효율을 좌우하는 큰 요소이다. 여기서 효율은 여러가지 방법으로 측정될 수 있으나, 이 논문에서는 Wiener 수[3, 4]를 사용한다. Wiener 수란 주어진 그래프 $G=(V(G),E(G))$ 의 모든 노드 간 거리의 합을 가리킨다. 그래프 내의 임의의 두 노드 $u, v \in V(G)$ 간 거리 $d_G(u, v)$ 는 두 노드 사이의 최단 거리로 정의되며 그래프의 Wiener 수는 다음과 같이 정의할 수 있다.

$$\sigma(G) = \frac{1}{2} \sum_{u \in V(G)} \sum_{v \in V(G)} d_G(u, v)$$

$\sigma(G)$ 은 모든 노드에서 다른 모든 노드까지의 거리의 합이므로 이 숫자가 적다는 것은 모든 노드들에서 다른 모든 노드들까지 통신의 거리가 짧다라는 것을 나타낸다. 따라서 노드들이 주어졌을 때 $\sigma(G)$ 의 값이 최소가 되는 트리인 최소 Wiener 수 신장트리로 네트워크 토폴로지를 구성하는 것이 가장 효율적인 네트워크 토폴로지라고 말할수 있다. 하지만 최소 Wiener 수 신장트리를 구하는 것은 NP-HARD 인 것이 증명되었다.[4]. 그나마 최소 Wiener 수 신장트리를 구하기 위한 분기 한정 알고리즘[3]이 있지만 이 또한 노드의 개수가 10 개를 넘어가는 경우 계산하는데 많은 시간이 소요되기에 이를 사용하여 최상의 네트워크를 구성하는 것은 어렵다. 하지만 이미 만들어진 네트워크에서 $\sigma(G)$ 값을 구할 수 있으므로, 따라서 DTG 로 생성된 네트워크 토폴로지의 효율성도 계산하여 비교할 수 있다.

3. 노드 degree 를 이용한 트리생성 알고리즘

그림 1 (a)의 노드들 중, 노드 d 의 degree 가 가장 큰 것을 알 수 있고, 따라서 d 에 최대한 많은 노드들을 연결시켜 만든 그림 1(d)가 2 장에서 설명한 기준에 의거하여 가장 효율적인 네트워크형태가 됨을 알 수 있다. 이 직관적 관찰을 일반화 시키면, degree 가 큰 노드에 가능한 한 많은 노드들을 연결하면 효율적인 네트워크 형태를 이룰수도 있다는 결론에 도달하며, 이장에서는 이에 기반한 DTG(degree-based Tree Generation) 알고리즘을 제안한다.

3.1. Degree-Based Tree Generation: DTG

이 논문에서 제안한 DTG 는 이웃한 노드들이 가지고 있는 연결선 degree 의 숫자를 비교하여 더 높은 쪽으로 트리형태를 만들어 감으로써 가능한 한 효율성이 높은 형태의 트리를 생성한다. DTG 는 모든 트리를 바탕으로 하는 라우팅에 적용될 수 있으나, 이해를 돕기 위해 AIR 를 사용하여 설명한다.

DTG 에서 각각의 노드는 1)자신이 이미 트리에 포함되었는지의 여부, 2)주변에 연결되었거나 연결이 가능한 노드의 숫자(degree)정보, 3)트리에 이미 포함되어 있다면 AIR 를 적용하여 자신에게 부여된 PL 등의 정보를 갖는다. 그림 2 를 참조하라.

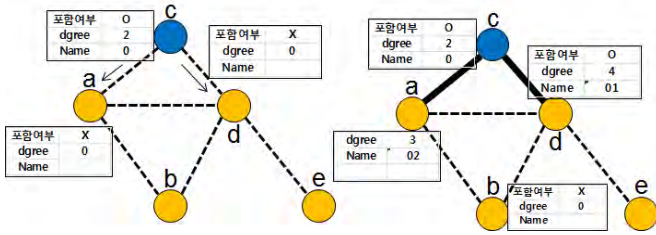
DTG 는 크게 두 단계로 형성되어 있는데, 1)루트노드를 시작으로, 노드들을 트리에 포함시키는 1 단계 (즉, 선으로 트리에 연결시키는 단계)(그림 2)와, 2)노드 degree 에 따라 노드간의 1 단계에서 형성된 기존 연결선을 변경하는 2 단계이다(그림 3). 예를들어 DTG 가 a 를 루트노드로 선택했다 가정하면, 루트노드 정보는 그림 2 에서 보듯이 트리포함여부, degree, PL 은 각각 0, 2, 0 을 갖는다(그림 2 의(A)). 다음, 루트 노드에 이웃되어 있는 노드들은, 즉 b 와 c 는, 루트노드의 하위노드로 연결선이 이어지고 각각 PL 01 과 02 를 부여받는다(그림 2 의 (B)). 이렇게 트리에 포함되면서 PL 을 부여받은 각각 노드들은 각자 다음단계를 진행한다. 먼저 노드는 자신과 연결이 되어있는 주변 노드들에게 위의 세가지 정보를 받아온다. 그래서 그 정보를 바탕으로 트리에 연결된 노드(예를 들어 b)는, 만약 기존 연결선으로 연결된 노드(즉 a) 보다 더 degree 가 높은 노드(즉 c)가 있다면 기존연결(즉 a 와의 선)을 끊고 가장 높은 degree 를 가진 노드(즉 c)의 하위 노드로 편입하며 선으로 연결한다. 이때 PL 을 통해 자신이 연결할 노드가 자신의 하위노드인지를 판별하여 하위노드일 경우에는 순환구조를 방지하기 위해 연결을 하지 않는다. b 는 c 의 하위노드가 아니므로 b 와 c 는 연결되고 b 는 PL 값으로 012 를 갖는다.(그림 3 의 (A)).

트리에 포함되지 않은 노드일 경우 자신의 하위 노드로 편입하여 PL 을 지정해주고 알고리즘을 마친다.

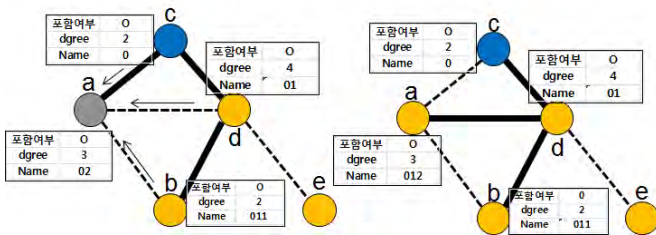
3.2. DTG 의 정확성 증명

AIR 는 기본적으로 트리 형태의 네트워크 토폴로지를 구성한다. 따라서 DTG 의 정확성을 증명하기 위해서는 완성된 네트워크 토폴로지가 트리 형태를 지니고 있는가, 즉 1) N 개의 노드에 대하여 N-1 개의 연결선을 가지며, 2) root 노드를 제외

한 모든 노드는 오직 한 개의 부모노드를 가지며,
3) 순환구조를 가지지 않는다는 특징을 증명하여야 한다.



(그림 2) DTG의 1 단계 - 노드를 트리로 포함



(그림 3) DTG의 2 단계 - 연결선의 재구성

DTG에서 연결선을 잇는 경우는 1) 노드들을 처음으로 트리에 포함시키는 경우와 2) 노드 degree에 따라 노드간의 기존 연결선을 바꾸는 경우 두가지이다. 첫번째의 경우 하나의 노드를 트리에 포함시킬 때 하나의 연결선을 생성한다. 따라서 i 개의 노드가 트리에 더 포함이 되더라도 연결선의 수도 똑같이 i 개가 늘어나므로 전체 노드 N 개에 대하여 연결선 $N-1$ 개의 숫자의 변화는 없다. 또한, 두번째 연결선을 바꿀 때에도 기존의 연결선을 없애고 새로운 연결선으로 연결하므로 총 연결선의 개수는 변화가 없다. 따라서 DTG로 생성된 트리는 노드 N 개에 대하여 연결선 $N-1$ 개를 지닌다. 또한, 새로운 노드를 연결할 때 항상 leaf 노드로 연결이 되기 때문에 각 노드는 오직 1개의 부모노드만 가진다. 마지막으로 트리에서 순환 구조라 함은 임의의 두 노드가 공동의 조상노드를 갖는 경우를 말하는데, DTG에서는 전달받은 PL의 정보를 통해 이를 쉽게 판별할 수 있어 방지할 수 있다.

4. 효율성 비교분석

4.1. 시뮬레이션 환경

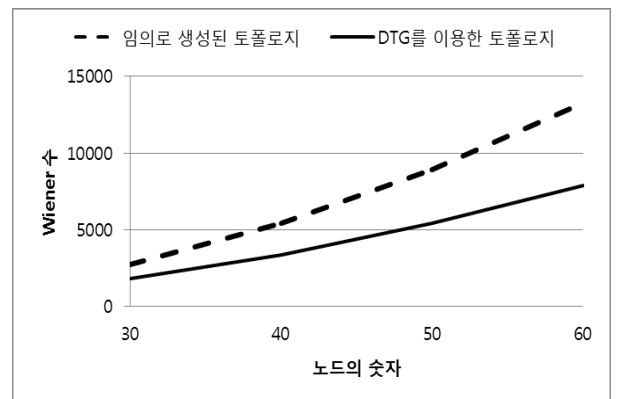
시뮬레이션은 가상 네트워크의 사이즈를 100×120 로 설정하고 노드 전송범위는 30으로 정하였다. 그 뒤 노드의 수가 각각 30, 40, 50, 60개인 경우를 설정하여 시험하였다. 그리고 각각의 경우

마다 임의로 형성된 네트워크 토폴로지의 Wiener 수와 DTG를 이용하여 생성된 네트워크 토폴로지의 Wiener 수를 1000번의 시뮬레이션하였다.

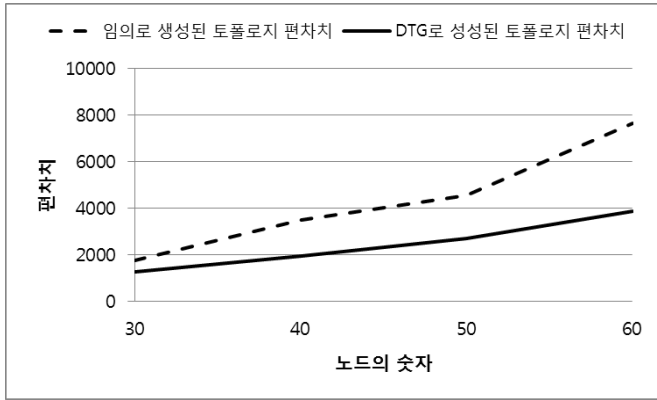
4.2. 시뮬레이션 결과

그림 4는 모든 시뮬레이션 결과에서 나온 Wiener 수를 평균을 내어 그래프로 나타내었다. 보는 바와 같이 DTG를 이용한 네트워크 토폴로지의 Wiener 수 값이 임의로 생성되는 네트워크 토폴로지의 Wiener 수 값보다 낮게 나온 것을 확인할 수 있다. 그 값은 노드의 숫자가 적을 때보다 노드의 숫자가 늘어날수록 차이가 점점 더 많이 나게 되는데 평균적으로 약 37% 가량 Wiener 수가 적게 나왔다. 따라서 DTG를 이용해 생성한 네트워크 토폴로지가 효율적임을 알 수 있다.

다음으로 Wiener 수의 편차치에 대하여 비교 분석해보았다. Wiener 수 편차치란 시뮬레이션 동안 생성된 네트워크 토폴로지의 Wiener 값의 최대값과 최소값의 범위를 말한다. 이러한 편차치가 적다는 것은 네트워크 트리에 변화가 있더라도 언제나 일정한 네트워크 성능을 얻을 수 있다는 이야기이다. 이것은 곧 네트워크의 신뢰성과 안정성에 영향을 준다. 그림 5는 임의로 생성된 토폴로지의 편차치와 DTG로 생성된 토폴로지의 편차치를 그래프로 나타낸 것이다. 그림 5의 그래프에서 보는 것 처럼 DTG를 이용하여 생성된 트리가 임의로 생성된 트리보다 편차치가 낮게 나온 것을 알 수 있다. 그 수치는 약 40% 가량으로 다시 말하자면 DTG가 임의로 생성된 토폴로지보다 40% 가량 신뢰성이 높다는 이야기이다.



(그림 4) Wiener 수 비교



(그림 5) 편차치 비교

5. 결론

본 논문은 서론에서 말한바와 같이 효율성 측정방법으로 Wiener 수를 제시하고 DTG 라는 새로운 트리 생성 알고리즘을 제안하였다. 그 목적은 트리형태의 네트워크 토폴로지를 이용하는 알고리즘에서 효율적인 토폴로지를 구성하는 것에 있다. DTG 의 핵심 아이디어는 이웃한 노드들과 degree 를 비교하여 degree 가 높은 노드로 가능한 많은 노드와 연결선을 연결하는 것이다. 시뮬레이션 결과 또한 DTG 가 Wiener 수치가 현저히 낮은 안정적인 형태의 트리구조를 가질 수 있는 토폴로지를 생성하는 것을 알 수 있다.

하지만 DTG 를 실제 라우팅에 사용하기에는 아직 해결해야 할 문제들이 남아있다. 먼저 루트노드를 정하는 알고리즘이 없기에 루트노드가 어디서 시작하는냐에 따라 PL 의 길이가 매우 길어질 수 있다. 길어진 PL 은 노드간 통신에서 많은 데이터공간을 차지하게 되 데이터 낭비로 이어질 수 있다. 따라서 가능한 루트노드를 네트워크 토폴로지에서 중앙에 위치하도록 하여 PL 을 줄일 수 있는 방법은 생각해 보아야 할 것이다. 다음으로 많은 노드가 좁은 공간에 밀집되어 있는 경우 하나의 노드로 연결선이 몰리는 경우가 생겨 하나의 노드에 과부하가 생기는 경우가 있다. 이 또한 적절한 방법으로 노드의 과부하를 줄일수 있는 방법을 찾아보아야 할 것이다. 또 아주 적은 경우이지만 DTG 를 이용해 만들어진 네트워크 토폴로지가 기대만큼 좋은 성능을 내지 못할 경우도 있다. 이러한 부분은 앞으로 연구를 통해 더욱 개선해야 할 점이 될 것이다.

참고문헌

[1] J.J. Garcia-Luna-Aceves and D. Sampath, "Scalable Integrated Routing Using Prefix Labels and Distributed Hash Tables for MANETs," *Proc. IEEE MASS 2009*, Oct 2009.

[2] R. Ghosh and J.J. Garcia-Luna-Aceves, "Automatic

Routing Using Multiple Prefix Labels," *Proc. IEEE GlobeCom 2012, Dec. 2012*.

[3] 강승호 김기영 이우영 송익호 정민아 이성로 (베이스 노드의 이동성이 큰 센서 네트워크 환경에서 최소 Wiener 수를 갖는 라우팅 트리를 위한 분기한정 알고리즘) *한국통신학회논문지 제35권 제5호(무선통신)*.

[4] M. Fischermann, et al. "Wiener index versus maximum degree in trees," *Discrete Applied Mathematics*, Vol.122, pp.127-137, 2002.

[5] Ghosh, R.; Garcia-Luna-Aceves, J.J. "Automatic incremental routing using multiple roots," *IEEE IPCCC*, 2013.