이동 Ad Hoc 네트워크에서 Threshold - Tuning을 통한 균형적인 에너지 소모 알고리즘

장재호O 장주욱
서강대학교 전자공학과
{jaehoO, jjang}@sogang.ac.kr

A Balanced Energy Consumption Algorithm by Threshold - Tuning for Mobile Ad Hoc Networks

Jaeho Chang0 Juwook Jang
Dept. of Electronic Engineering, Sogang University

요 약

Ad Hoc 네트워크에서 Threshold - Tuning을 통한 노드들간의 균형적인 에너지 소모를 위한 알고리즘(BECT : A Balanced Energy Consumption Algorithm by Threshold - Tuning)을 제안한다. BECT는 노드간 에너지 균형을 맞추어 네트워크의 수명(Network Lifetime)을 연장한다. 제안한 알고리즘은 DSR(Dynamic Source Routing) 프로토콜을 기반으로 구현하였으며, GloMoSim 2.0을 이용하여 실험을 하였다. 실험 결과 BECT가 실험 토폴로지에 따라 DSR의 에너지 균형을 17 - 31% 향상시키며, 데이터 전송율이나 제어 패킷 비율에 있어서도 향상된 성능을 보여준다.

1. 서 론

이동 Ad Hoc 네트워크에서 배터리로 동작하는 노드들은 에너지가 제한되어 있기 때문에, 최근 MAC, 네트워크 프로토콜, 또는 응용 계층에 걸쳐서 에너지 또는 전력 절감을 목표로 한 연구들이 진행되었다 [2,3,5,6,7]. 기존 Ad Hoc 라우팅 프로토콜은 최단 경로를 통해 데이터 패킷을 송수신한다. 이 경우 일부 노드들에 패킷의 흐름이 집중되면 이 노드들은 다른 노드들에 비하여 에너지 소모가 심하게 되고, 따라서 전체 네트워크의 에너지 불균형이 초래된다. 본 논문에서는 네트워크 계층에서 에너지 소모의 균형을 통하여 전체 네트워크 수명(lifetime)을 증가시키기 위한 알고리즘을 제안한다. 즉 라우팅 경로를 설정함에 있어 에너지가 어느정도 이상 되는 노드들로 이루어진 경로를 선택함으로써 전체 네트워크의 수명을 연장함을 목표로 한다. 이와 관련하여 에너지가 충분한 라우팅 경로 설정과 설정된 경로를 통한 데이터 전송중 경로의 에너지가 부족해지는 경우 새로운 경로를 찾아 목적지 노드까지 경로를 유지해 주는 기능을 간단하고 효과적인 Threshold - Tuning 기법을 이용한 알고리즘(BECT : A Balanced Energy Consumption Algorithm by Threshold - Tuning)을 제안한다. 2장에서는 Ad Hoc 라우팅 프로콜인 DSR(Dynamic Routing Protocol)[1]에 대해 알아보고, 3장에서는 BECT를 소개하며, 4장에서는 시뮬레이션 결과와 분석하고, 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. Ad Hoc 라우팅 프로토콜 (DSR)

DSR(Dynamic Routing Protocol)[1]은 Ad Hoc 라우팅 프로토콜 중 On - demand 방식으로 동작하는 프로토콜이다. 즉 소스에서 목적지 노드로 보낼 데이터 패킷이 있을 경우 라우팅 캐쉬를 검색하고 라우팅 캐쉬에서 목적지 노드로의 경로가 없을 경우 RREQ(Route Request)를 통하여 경로를 찾는다. 목적지 노드는 RREQ를 받았을 경우 RREP(Route Reply)를 소스로 보내며, 데이터 전송 도중 노드의 이동성으로 인하여 경로가 유지되지 않으면, 이를 발견한 중간 노드에서 RERR(Route Error)를 소스 노드에게 보내주어 새로운 경로를 찾도록 한다.

2.1. DSR의 라우팅 방법

DSR은 라우팅 경로 설정에 있어서 최단 경로, 즉 목적지 노드까지 홉(hop) 수가 가장 적을 경로를 선택하며, 에너지를 메트릭(metric)으로 하지 않는다. 따라서 선택한 경로가 유효하지 않을 때까지 그 경로만을 이용한다. 그림 1에서 N2, N6가 에너지가 얼마 남지 않은 노드라고 가정할 때, N1 - N3 - N4 - N5 - N8 또는 N1 - N3 - N4 - N7 - N8을 선택함이 에너지 균형을 이루는 경로 선택이라고 할 수 있는데, DSR은 최단 경로인 N1 - N2 - N5 - N8을 선택한다.

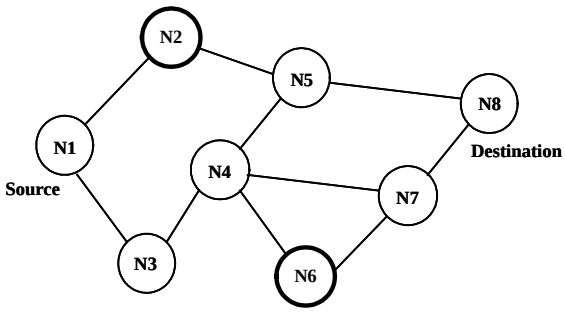

그림 1. Ad Hoc 라우팅의 예

그림 2에서 DSR은 최단 경로인 N1 - N3 - N2을 선택하여 N3가 에너지를 다하거나 N3로의 링크가 유지되지 않을 때까지 이 경로를 이용한다. 이 경우 N3의 에너지가 다른 노드에 비하여 현저히 줄어들게 되고 에너지의 불균형이 초래된다.

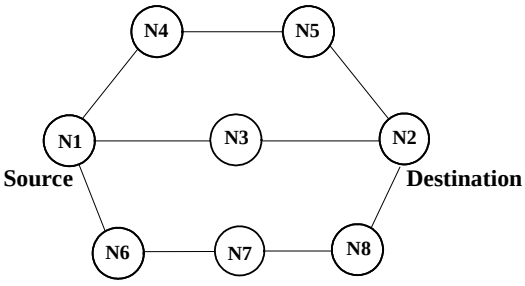

그림 2. Ad Hoc 라우팅의 예

3. 에너지 균형 알고리즘 (BECT)

에너지가 부족한 노드들이 라우팅 경로에 포함되어 전체 네트워크의 수명을 단축시키는 문제점을 해결하기 위하여 Threshold - Tuning을 통한 균형적인 에너지 소모 알고리즘 (BECT : A Balanced Energy Consumption Algorithm by Threshold - Tuning)을 제안한다. BECT는 DSR을 기반으로 하여 몇가지 파라미터와 간단한 기능의 추가로 구현한다.

3.1. BECT의 파라미터

모든 노드는 파라미터로서 에너지 Threshold(*ethn*), 남은 에너지 레벨(*rel*), 에너지 Threshold 감소량 (*eth_dec*)을 유지하고 있는다. *ethn*은 Threshold - Tuning 메카니즘에 의해 조절되며(3.2 장), *eth_dec*는 3.3장 및 3.4장에서 논의한다. RREQ, RREP, RERR와 데이터 패킷을 포함한 모든 DSR 패킷에는 8 bits의 예약된 부분(reserved field)이 있다[4]. 새로운 타입의 제어 패킷을 추가 하지 않고 이 부분에 패킷의 에너지 Threshold(*ethp*)을 추가하여 Threshold Tuning과 경로 설정 및 경로 변경에 이용한다. *ethn*은 각 노드에 추가되며, *ethp*는 패킷에 추가되는 변수이다.

3.2. Threshold - Tuning 메카니즘

소스 노드가 RREQ 패킷을 생성함에 있어 패킷의 *ethp*를 자신의 *ethn*으로 세팅하여 전송한다. 중간 노드들이 패킷을 수신시 패킷의 *ehtp*와 자신의 *ethn*을 비교하여 *ethp*가 *ethn*보다 작으면 자신의 *ethn*을 *ethp*값으로 조절한다. DSR은 RREQ, RREP, RERR의 주요한 제어 패킷을 사용하는데 이중 RREQ는 브로드캐스트로 전체 네트워크로 퍼지는 특징이 있다. 따라서 RREQ를 포함한 모든 DSR 패킷에 *ethp*를 넣어 전체 네트워크의 Threshold를 Tuning한다. Threshold Tuning의 효과는 4장의 실험 결과를 통하여 검증된다. Ad Hoc 네트워크의 노드들이 Promiscuous 모드로 동작할 경우 overhear하는 모든 패킷, 즉 자신이 경로에 포함되어 있지 않아도 수신하는 패킷의 *ethp*로부터 자신의 *ethn*을 조절 할 수 있으므로 효과는 더 커진다.

3.3. BECT의 경로설정

소스 노드가 RREQ 패킷을 생성함에 있어 패킷의 *ethp*를 자신의 *ethn*으로 세팅하고, 중간 라우터들은 RREQ를 받으면 자신의 에너지 레벨 *rel*과 패킷의 *ethp*를 비교하여 라우팅에 참여 여부를 결정한다. 즉 *rel*이 *ethp*보다 크면 기존 DSR과 같이 RREQ를 브로드캐스트하고, 작으면 패킷을 버림으로써 라우팅에 참여하지 않도록 한다. 만약 에너지가 충분한 경로가 발견되지 않을 경우 소스 노드는 *ethn*을 *eth_dec* 만큼 낮추어 다시 RREQ를 시행한다.

예를 들어 그림 3에서 N2, N6의 에너지 레벨(*rel*)이 *ethp*보다 작을 경우 라우팅 경로는 N1 - N2 - N5 - N8 대신, N1 - N3 - N4 - N5 - N8 또는 N1 - N3 - N4 - N7 - N8이 선택된다.

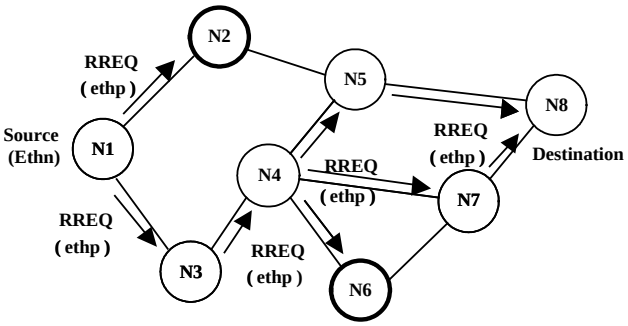

그림 3. BECT의 경로 설정 과정

3.4. 중간 노드의 데이터 전달

소스 노드에서 모든 데이터 패킷의 *ethp*부분에 *ethn*을 넣어 전송하고, 중간 노드에서 데이터를 전달할 때 *ethp*와 자신의 *rel*을 비교하여 *rel*이 *ethp*보다 작을 경우 RERR을 소스에게 보낸다. 소스에서는 이 경로를 라우트 캐쉬에서 지우고 캐쉬에 있는 다른 경로를 이용하거나 *ethn*을 *eth_dec*만큼 낮추어 새로 경로를 찾는다(3.3장). 그림 2에서 최초 N1 - N3 - N2의 경로를 선택하여 데이터를 전송하다가, N3의 에너지 레벨(*rel*)이 *ethp*보다 작으면 N1 - N4 - N5, N1 - N6 - N7 - N8의 경로를 순차적으로 이용한다.

4. 실험 및 분석

우선 간단한 수학적 분석을 통하여 BECT의 이론적인 에너지 균형에 대해 분석해 보고, 시뮬레이션을 통하여 BECT의 실제 성능 및 효과를 실험한다.

4.1. 이론적인 BECT의 행동 성능 분석

그림 2의 토폴로지에서 BECT는 전술한 바와 같이 계속적으로 세가지 경로를 바꾸어 가면서 사용한다. 이때 경로를 변경하는 빈도(frequency)는 *eth_dec*에 의해 결정되는데 한 경로를 선택한 시간(Δt)은 다음과 같은 간단한 방법으로 계산할 수 있다.

우선 노드가 대기(Idle 혹은 Listening) 모드와 수신(Receiving) 모드에 있을 때 에너지 소모는 같고, BECT의 제어 패킷에 의한 에너지 소모는 무시할 수 있다고 가정한다. 이때 유효 대역폭을 B bps, 모든 헤더를 포함한 데이터 패킷의 크기를 k bits라 하고, 송수신 파워를 각각 P_{tx}, P_{rx}라 하면 패킷으로 송수신 할 때 사용되는 에너지는 각각 다음과 같다.

$$W_{tx} = \frac{P_{tx} \times k}{B \times 3600} \quad (1), \quad W_{rx} = \frac{P_{rx} \times k}{B \times 3600} \quad (2)$$

그림 1에서 모든 노드의 초기 에너지양은 같다고 하면, DSR과 BECT를 사용하였을 경우 시간에 따른 경로 선택은 그림 4와 같다. 소스 노드에서 초당 n개의 데이터 패킷을 송신한다고 하면, 이론적인 BECT의 하나의 경로 선택 시간(Δt)은 다음과 같이 구할 수 있다. 식 (3)은 노드가 non - promiscuous 모드로 동작 할 경우 Δt 이고, 식 (4)는 promiscuous 모드로 동작할 경우의 Δt 이다.

$$\Delta t_n = \frac{eth\_dec}{n \times (W_{tx} + W_{rx})} \quad (3)$$

$$\Delta t_p = \frac{eth\_dec}{n \times (W_{tx} - W_{rx}) + P_{rx} \Delta t_p} \quad (4)$$

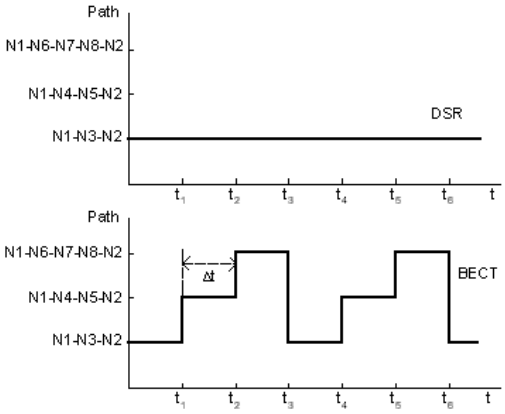

그림 4. DSR과 BECT의 경로 설정

4.2, 4.3장에서는 좀더 구체적인 알고리즘의 효과을 살펴보기 위하여 시뮬레이션을 통해 DSR과 BECT의 성능을 비교한다.

4.2. 실험 환경

GloMoSim 2.0[4]을 이용하여 실험을 하며, 1000m*1000m 구역에 50개의 노드를 균등 분포시키고, 각 노드는 random waypoint 모델에 의하여 속도 0 - 1m/s의 이동성을 주었으며, 정지시간(pause time) 동안 먼추었다가 이동을 계속한다. 10개의 소스노드가 초당 5개의 CBR 트래픽을 발생시키며, 패킷의 크기는 512 bytes로 네트워크의 용량에 비해 충분히 작은 값을 사용한다. 에너지 모델은 Lucent 2Mb/s WaveLAN 802.11 랜카드에 기반하였으며[5], 실험에서 idle 모드에서의 에너지 소모는 무시할 수 있고[3], 노드가 non - promiscuous 모드로 동작한다고 가정한다.

4.3. BECT의 에너지 균형 및 성능

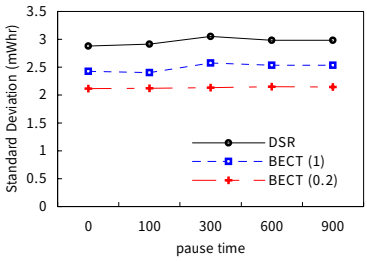

그림 5. Standard Deviation

에너지 균형 정도를 알아보기 위하여 900초 시뮬레이션 후 각 노드의 남은 에너지의 표준 편차를 이용한다. 그림 5는 DSR과 BECT의 *eth_dec*의 값을 0.2와 1mWhr로 하였을 때의 결과이다. DSR에 비하여 BECT(0.2, 1)은 각각 17, 31% 성능의 향상을 보인다. 데이터 전달율은 그림 6과 같이 DSR과 BECT 모두 95%이상을 보이는데 BECT의 전달율이 더 높음을 알 수 있다. 이는 모바일 환경의 Ad Hoc 네트워크에서 링크 브레이크에 의하여 패킷이 손실되는 경우에 있어서 BECT의 경우 에너지 레벨에 의한 사전 경로 변경을 통하여 높은 전달율을 보인 것으로 볼 수 있다. 그림 7은 전체 패킷 중 제어 패킷의 비율인데 위와 같은 이유로 BECT를 적용한 경우 RERR과 RREQ의 발생이 적어 오히려 제어 패킷 또한 적어지는 효과를 보인다.

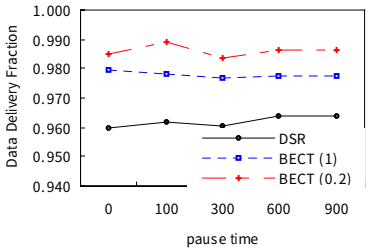

그림 6. Data Delivery Fraction

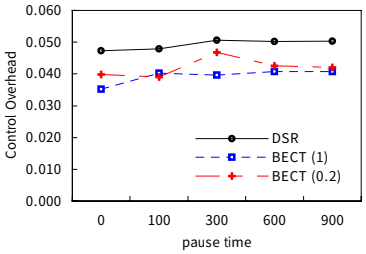

그림 7. Control Overhead Fraction

5. 결 론

본 논문에서는 Ad Hoc 라우팅에 있어서 노드간 균형된 에너지 소모를 위한 알고리즘을 제안하였다. BECT는 DSR에 간단하고 쉽게 구현할 수 있으며, 에너지 소모의 균형 이외에 데이터 전송율이나 제어 패킷 비율에 있어서도 향상된 성능을 보여준다. 향후 BECT의 파라미터에 의한 성능 개선 및 다른 에너지 균형 알고리즘과의 비교 분석을 하고자 한다.

6. 참고 문헌

[1] David B. Johnson, David A. Maltz, Yih-Chun Hu , and Jorjeta G. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). IETF Internet Draft.

[2] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed Energy Conservation for Ad Hoc Routing. In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pp. 70-84., ACM. July, 2001.

[3] Hagen Woesner, Jean-Pierre Ebert, Morten Schlager, Adam Wolisz. Power Saving Mechanisms in Emerging Standards for Wireless LANs: the MAC Level Perspective. IEEE Personal Communications, Vol. 5, Issue 3, pages 40-48, Jun. 1998.

[4] Xiang Zeng, Rajive Bagrodia, Mario Gerla. GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks. Parallel and Distributed Simulations Conference (PADS), 1998.

[5] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, Robert Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. To appear in ACM Wireless Networks Journal, Volume 8, Number 5, Sep. 2002.

[6] Mike Woo, Suresh Singh, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. International Conference on Mobile Computing and Networking (MobiCom '98), pages 181-190, Oct. 1998.

[7] Kyungtae Woo, Chansu Yu, and Dongman Lee. Non-Blocking, Localized Routing Algorithm for Balanced Energy Consumption in Mobile Ad Hoc Networks. International Symp on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2001), Aug. 2001.