

지그비 패킷을 위한 최적의 패킷유지시간 설정 방법

A Suitable Packet Time-To-Live Value for ZigBee Packets

이광국, 전영호, 신진규, 박홍성

Kwang Koog Lee, Yeong Ho Jeon, Jin Gyu Shin, Hong Seong Park

Abstract - These days wireless sensor networks receive much attention from industry and researchers on various fields. The challenge is that wireless sensor networks have limited resources. Nevertheless, the route discovery in ZigBee sensor networks is performed by simple flooding when the original packet is rebroadcasted at least once by every node in a network. Such uncontrolled flooding generates an excessive number of packets competing for the media and causes a high collision rate. In this article, we propose a suitable packet Time-To-Live value to solve problems of uncontrolled flooding in ZigBee networks. It is shown that more sufficient route discovery in a ZigBee network can save network resources and lengthen the life of a sensor network.

Key Words : ZigBee, Route Discovery, Radius, Packet Time-To-Live

1. 서론

유비쿼터스 환경에서의 무선 센서 네트워크는 스마트 센서와 무선 네트워킹 기술의 발전으로 많은 산업적 수요를 창출하고 있으며 그 연구 분야 역시 다양해지고 있다[1]. 이러한 무선 센서 네트워크는 응용에 따라 수십~수천 개의 소형 센서 노드들로 배치되기 때문에 제한된 배터리 용량을 가지고 동작하는 것이 일반적이다. 무선 센서 네트워크는 지리 및 환경적 요인에 많은 제한을 받기 때문에 경로 유실에 의한 경로 탐색 오퍼레이션이 자주 발생한다. 이러한 오퍼레이션은 기본적으로 플러딩에 기반을 두기 때문에 네트워크에 발생하는 수많은 브로드캐스팅으로 인한 패킷의 경쟁 및 충돌 문제를 발생시킨다[3]. 게다가 이러한 오퍼레이션은 수신된 패킷의 처리와 재전송을 위해 센서에게 많은 전력을 요구하기 때문에 센서의 수명을 단축시키는 원인이 된다. 결론적으로 비효율적인 플러딩을 제한함으로써 네트워크의 수명을 증가시키기 위한 연구가 뒷받침되어야 한다.

무선 상에서 이러한 비효율적인 플러딩을 막기 위해 이미 많은 기법들이 제안되었으며, 크게 세 가지로 그 유형을 분류할 수 있다. 첫 번째는 플러딩에 의한 경쟁 및 충돌을 피하기 위해 전송되는 패킷을 스케줄링 하는 기법이다[3]. 이 기법에서는 브로드 캐스팅되는 패킷에 대하여 랜덤 지연을 주어 다시 브로드 캐스팅되는 시간을 다르게 하여 충돌과 경쟁을 피하는 구조를 취한다. 두 번째는 위치기반에 의한 정보를 통해 플러딩을 제한하는 기법이다[4]. 이 기법에서는 GPS와 같은 장치를 사용하여 노드의 위치 정보를 알고 브로드캐스팅을 제한하거나 지역적으로 브로드캐스팅을 수행하도록 유도한다. 마지막으로 세 번째 기법은 확률적인 계산을 통해 브로드캐스팅의 범위를 한정하는 방법이다[5]. 이 기법에서는 브로드캐스팅 되는 영역을 홉 수 또는 거리로 한정하여 브로드캐스팅을 필요로 하지 않는 지역으로의 비효율적인 플러딩

현상을 억제한다.

본 논문에서는 세 번째 기법을 기반으로 하여 센서 네트워크의 한 예로 들 수 있는 지그비(ZigBee) 네트워크에서 플러딩을 제한하기 위한 최적의 패킷유지시간을 설정하는 방법을 제안한다. 지그비는 근거리 통신을 지원하는 저속, 저 전력 지향의 유비쿼터스 컴퓨팅을 위한 표준 기술 중의 하나로써 지그비 센서를 통하여 무선 매쉬 네트워크를 구축할 수 있는 가능성을 언급하고 있다[2,7]. 지그비에서는 이러한 매쉬 네트워크의 구축을 위해 프로토콜 스택 상의 네트워크 계층에서 경로탐색에 기반을 둔 최적 경로의 라우팅 방법을 제공하는데, 이 때 지그비는 목적지 간의 데이터 전송 및 경로 확립을 위한 라우팅 패킷에 대하여 패킷유지시간을 설정해야 한다. 지그비의 경로 탐색은 AODV(Ad hoc On-demand Distance Vector) 경로 탐색과 같은 플러딩에 기반을 두기 때문에 기본적으로 패킷유지시간은 패킷이 지그비 네트워크 모든 곳에 접근될 수 있는 최대값으로 설정된다[6]. 따라서 과도한 라우팅 패킷으로 인한 네트워크 리소스 낭비 및 센서 수명의 단축을 막기 위해서 최적의 패킷유지시간이 요구된다.

기본적으로 지그비는 계층적인 어드레싱 방식을 통해 계층적 트리를 형성할 수 있기 때문에 계층적 트리 라우팅을 통한 홉 수를 최적의 경로를 위한 홉 수의 상한으로 설정할 수 있다. 결국 이러한 홉 수의 상한을 패킷유지시간으로 적용하면 경로 탐색 시 불필요한 노드의 경로 참여를 줄일 수 있기 때문에 네트워크의 성능을 높이고 지그비 디바이스의 필요 없는 에너지 소모를 줄일 수 있다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 제안된 방법을 뒷받침하기 위한 지그비 네트워크 계층의 어드레싱 방법에 대하여 알아보고, 3장에서는 본 논문에서 제안하는 패킷유지시간 설정 방법에 대하여 살펴본다. 4장에서는 제안된 방법을 시뮬레이션을 통하여 평가 및 분석하고 마지막으로 5

장에서 결론을 맺는다.

2. 지그비 노드의 타입과 어드레스

지그비 노드는 코디네이터(PAN Coordinator), 라우터(Router), 종단 디바이스(End Device)의 세 가지 타입으로 분류된다[7]. 코디네이터 및 라우터는 지그비 네트워크의 새로운 PAN(Personal Area Network)을 형성하거나 멀티홉 데이터 전송을 위한 라우터 기능을 갖고 네트워크 계층에서 설정되는 네트워크 환경 설정 파라미터를 기반으로 자신의 지식으로 새롭게 참여하는 노드들에 대하여 새롭게 주소를 할당하거나 최적 경로를 설정하기 위한 경로 탐색 과정에 참여하는 노드를 말한다. 이와 달리 종단 디바이스는 코디네이터 및 라우터의 기능을 최소화한 노드으로써 주소를 할당하는 능력이 없을 뿐만 아니라 멀티홉 전송을 위한 경로 탐색 과정에 참여하지 못하고 데이터의 멀티홉 전송을 지원하지 못하기 때문에 라우팅 과정에서 중간노드의 역할을 수행하지 못한다.

지그비 네트워크에서 각 노드의 주소는 16비트 Short Address를 갖는다. 코디네이터의 경우 지그비 네트워크의 형성자이므로 항상 16진수 값인 0x0000의 주소를 가지며 이후 참여하는 노드들에 대하여 코디네이터와 종단 디바이스 별로 각각 수식화된 주소 공식에 의하여 계층적으로 주소를 할당하게 된다. 우선 코디네이터 및 라우터 이하 라우터로써 참여하는 자식 노드들은 부모 노드로부터 다수의 16비트 Short Address를 담고 있는 주소 블록(Address Block)을 할당받는다. 상기 주소 블록을 할당받은 자식 노드는 상기 주소 블록 상에서 최우선한 주소 값을 자신의 16비트 Short Address로 설정한다. 상기 주소 블록을 구하기 위해서는 우선 부모 노드들이 가지고 있는 자신의 깊이(d) 값에 대해서 할당할 수 있는 자식의 최대 개수를 나타내는 CSkip(d) 값이 필요하다. CSkip(d) 값은 자식의 최대 라우터 개수(R_m), 최대의 자식 노드 수(C_m), 최대 자식의 깊이(L_m) 등의 네트워크 파라미터 설정 값에 따라 수식 1의 공식에 의해 계산될 수 있다.

$$CSkip(d) = \frac{1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}}{1 - R_m}$$

수식 1 CSkip(d) 계산 공식

코디네이터 및 라우터 이하 종단 디바이스로 참여하는 자식 노드들은 부모 노드로부터 수식 2에 의한 하나의 16비트 Short Address를 할당 받는다. 수식 2에서 n은 주소를 할당하는 부모 입장에서 n번째로 참여하는 종단 디바이스를 의미한다.

$$A_n = A_{parent} + CSkip(d) \times R_m + n$$

수식 2 종단 디바이스 주소 할당 공식

3. 최적의 패킷유지시간 설정 방법

소스에서 목적지 노드까지 최적의 패킷유지시간을 설정하기 위해서는 트리 라우팅을 통해 연결될 수 있는 홉 수를 계산할 수 있어야 한다. 만일 경로 탐색을 통해 확립된 경로가 트리 라우팅을 통해 전송할 수 있는 홉 수보다 더 많은 홉 수를 필요로 한다면 경로 탐색은 의미가 없기 때문에 이와

같은 트리 라우팅을 통해 연결될 수 있는 홉 수를 경로 탐색 시 라우팅 패킷을 위해 필요한 최대 패킷유지시간으로 설정할 수 있다. 결국 최적의 패킷유지시간은 아래와 같은 수식 3에 의해 계산될 수 있다.

$$\max TTL = D_s + D_d - 2 \times B_s$$

수식 3 최대 TTL 설정 공식

수식 3에 의하면 maxTTL은 최적의 패킷유지시간을 의미한다. D_s는 코디네이터에서 소스까지의 깊이를 나타내고, D_d는 코디네이터에서 목적지까지의 깊이를 나타낸다. 마지막으로 B_s는 트리 구조상으로 소스와 목적지 간의 중첩되는 가지의 개수를 나타낸다. 여기서 D_s는 이미 알고 있기 때문에 D_d와 B_s만을 계산하면 된다. 지그비 노드의 경우 코디네이터에 의해 깊이 값에 따라서 계층적인 주소공식을 통해 주소를 할당받기 때문에 목적지 노드가 속한 주소를 현재 R_m, C_m, L_m과 같은 네트워크 파라미터에 의해 정의된 지그비 네트워크에서 계산될 수 있는 주소의 범위들과 비교를 통해 목적지 노드의 깊이를 알아낼 수 있다. 아래 그림 1은 현재 소스의 깊이 값에 대하여 목적지 노드의 깊이 값을 고려한 최적의 TTL 값을 설정하기 위한 알고리즘에 대한 흐름도를 나타낸 것이다.

그림 1에 따르면 소스 노드는 SrcDepth (소스 노드의 깊이), SrcAddr (소스 노드의 16비트 Short Address), DstAddr (목적지 노드의 16비트 Short Address)를 입력으로 전달한다. 다음으로 목적지 노드의 깊이를 계산하기 위해 CurDepth (현재 깊이)라는 임시 변수를 정의하고 코디네이터 이하 노드들로부터 깊이 탐색이 수행되므로 값을 1로 초기화한다. 지그비 노드의 경우 라우터의 주소는 주소 블록에 의해 범위가 한정되므로 깊이 값에 따라 목적지 주소를 결정하기 위해 라우터가 가질 수 있는 첫 번째 주소를 의미하는 LowerBound와 라우터가 가질 수 있는 최상위 주소를 의미하는 UpperBound라는 임시 변수를 정의한다. UpperBound 값은 현재 주소를 비교중인 라우터의 LowerBound 값에 현재 깊이에 따른 CSkip 값을 더하여 설정할 수 있다. 여기서 CSkip 값은 부모 입장에서 자식이 되는 목적지 노드의 주소의 범위를 나타내므로 현재 탐색중인 깊이 값에서 1을 감소한 값을 계산해야 한다. 예를 들면 깊이 1에서 UpperBound에 주소를 설정하기 위해서는 깊이 0의 CSkip 값이 필요하다. 다음으로 소스와 목적지 노드와의 중첩되는 가지의 개수를 계산하기 위하여 소스 노드의 주소가 LowerBound와 UpperBound 범위에 존재하는지 판단하여 만일 소스가 목적지 노드와 같은 가지 부모선상에 놓여있을 때 OverlappedBranch 변수를 1 증가한다. 다음으로 실제 목적지 노드의 깊이를 판단하기 위해 위 과정에서 설정된 목적지 주소가 LowerBound와 UpperBound 사이에 존재하는지 판단한다. 이때 목적지 노드의 주소가 현재 설정된 LowerBound와 UpperBound 사이에 존재하지 않는 경우 RouterIndex를 1증가하고 현재 LowerBound에 설정된 UpperBound 값을 저장하여 다음 라우터의 주소가 되는 범위와 비교를 다시 시작한다. 이 과정은 현재 지그비 네트워크에 설정된 라우터의 개수만큼 반복된다. 만일 라우터의 개수만큼 반복을 수행한 이후에도 목적지 주소가 발견되지 않

은 경우는 목적지 주소가 현재 깊이에서 중단 디바이스로써 활동하고 있음을 의미하므로 현재 깊이에서 비교 과정을 빠져나온다. 만일 위의 목적지 주소와 *LowerBound* 및 *UpperBound*의 비교에서 목적지 노드의 주소가 현재 설정된 *LowerBound*와 일치하는 경우 현재 탐색중인 깊이를 나타내는 *CurDepth*는 지금 현재의 목적지 노드의 깊이를 나타내므로 중단 디바이스를 위한 과정과 마찬가지로 비교 과정을 빠져나와 현재까지 저장된 *OverlappedBranch* 변수와 *SrcDepth* 변수를 수식 1을 사용하여 *maxTTL* 값을 계산한다. 만일 위의 목적지 노드 비교 과정에서 *LowerBound*와 목적지 주소가 일치하지 않으면서 *UpperBound*에 속하는 경우는 현재 깊이 이하의 임의의 깊이 내에 목적지 주소가 속했음을 의미하기 때문에 현재 깊이 *CurDepth*를 1 증가하고 증가된 깊이에서 *RouterIndex*를 다시 초기화하고 위와 같은 과정을 반복한다.

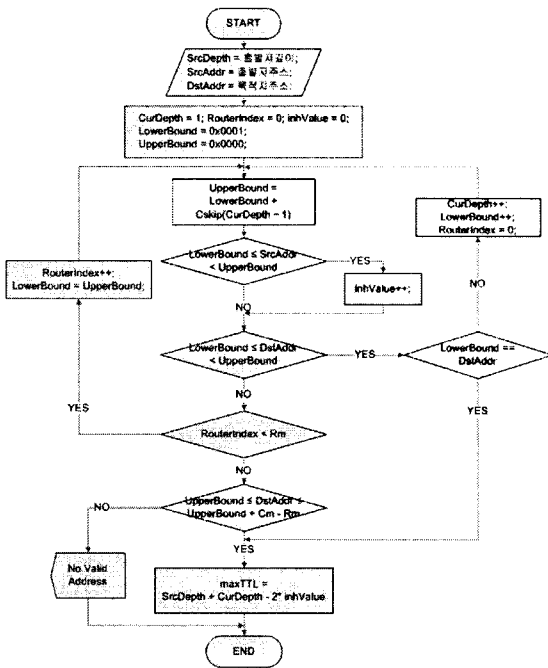


그림 1 패킷유지시간 설정 알고리즘

4. 성능 평가 및 분석

4.1 시뮬레이션 시나리오

본 논문에서는 제안한 알고리즘의 성능을 측정하기 위해 NS2 시뮬레이터를 사용하였다[8]. 시뮬레이션 측정을 위해 정방형 100 X 100의 평면적인 고정 센서 망을 구성하여 각 시나리오마다 200초 동안 0.70 pkts/sec의 packet rate로 CBR(Constant Bit Rate) 데이터를 전송하였다. 실제로 플러딩이 네트워크에 주는 영향을 측정하기 위해 노드의 개수를 20, 30, 40, 50, 60개의 순서로 증가시켰고 각 시뮬레이션 시나리오마다 10개의 트래픽 연결을 확립하였다. 제안된 최적의 패킷유지시간 설정 방법과 기존의 패킷유지설정 방법의 라우팅 오버헤드의 비교를 통해 성능을 비교하였다.

4.2 라우팅 오버헤드

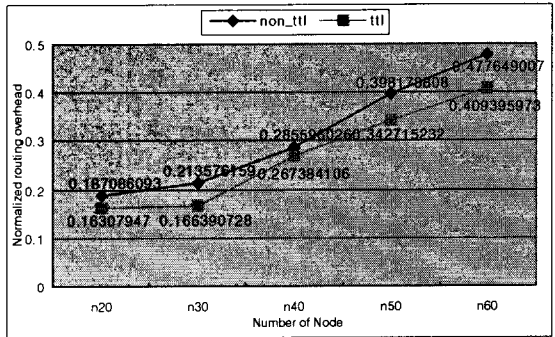


그림 2 노드 개수에 따른 라우팅 오버헤드

5. 결론

본 논문에서는 네트워크 계층상의 지그비 패킷에 대하여 최적의 패킷유지시간을 설정하기 위해 목적지 노드의 깊이를 소스 노드에서 미리 계산하여 경로 탐색 시 소요되는 비용의 상한을 결정하였다. 지그비의 경우 트리 라우팅을 통해 최적의 경로의 상한을 설정할 수 있기 때문에 플러딩의 범위를 제한할 수 있다. 제안된 방법은 시뮬레이션을 통해 확인한 결과 기존의 방법에 비해 네트워크의 라우팅 오버헤드를 낮출 수 있었다. 앞으로의 연구에서는 최적의 패킷유지시간의 설정을 수식화를 통한 이론적 방법을 통해 시뮬레이션 결과를 뒷받침할 수 있는 보다 구체적인 모델링 연구가 되어야 할 것이다.

참고 문헌

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, Aug. 2002.
- [2] ZigBee Alliance, "ZigBee specification : ZigBee document 053474r06 Version 1.0", 14 Dec. 2004.
- [3] S.Y. Ni, and Y.C.Tseng, "The Broadcast Storm Problem in a Mobile Ad Hoc Network", ACM, Mobicom 1999.
- [4] Ya Xu, J. Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing", Proc. ACM/IEEE, International Conference on Mobile Computing and Networking, 2001.
- [5] Z. Cheng and W.B. Heinzelman, "Flooding Strategy for Target Discovery in Wireless Networks", ACM, Mobicom 1999.
- [6] C.E Perkins and E.M. Belding-Royer and S.R. Das, "Ad hoc On-demand Distance-Vector (AODV) routing", IETF Internet Draft, draft-ietf-manet-aodv-10.txt, March 2002, Work in progress.
- [7] Patrick Kinney "Zigbee Technology : Wireless Control that Simply Works", Communication Design Conference, 2 Oct. 2003.
- [8] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>