
방송 패킷을 활용한 무선 애드혹 네트워크의 이웃 정보 전송 절감

최 선 응*

Neighbor Knowledge Exchange Reduction using Broadcast Packet in Wireless
Ad hoc Networks

Sun-Woong Choi*

본 연구는 2007년도 국민대학교 신진교수 연구지원금과 지식경제부 및 정보통신연구진흥원의
IT성장동력기술개발사업의 일환으로 수행되었음 [2008-S-041-01, u-City용 센서네트워크 PHY/MAC 개발]

요 약

무선 애드혹 네트워크에서 동작하는 많은 프로토콜들은 이웃 노드에 대한 정보를 중요하게 사용한다. 이웃 노드에 대한 정보는 Hello 메시지를 주고받음으로써 알 수 있다. 자신의 존재를 알리고 싶은 노드는 주기적으로 Hello 패킷을 전송한다. 하지만, 이러한 Hello 패킷의 전송은 무선 애드혹 네트워크에 큰 제어 부하로 작용하게 된다. 이 논문에서는 Hello 패킷뿐만 아니라 방송 패킷을 활용하여 이웃 노드 정보를 주고받는 방법을 고찰한다. 분석과 컴퓨터 시뮬레이션을 통하여 방송 패킷을 활용하는 기법이 Hello 패킷만을 사용하는 방식에 비하여 상당한 효과를 보인다는 것을 분석한다. Hello 패킷의 전송 주기와 방송 패킷의 전송 주기가 같다면 42% 정도의 부하 절감 효과가 있다.

ABSTRACT

Neighbor knowledge in wireless ad hoc networks provides important functionality for a number of protocols. The neighbor knowledge is acquired via Hello packets. Hello packets are periodically broadcasted by the nodes which want to advertise their existence. Usage of periodic Hello packet, however, is a big burden on the wireless ad hoc networks. This paper deals with an approach where each node acquires neighbor knowledge by observing not only Hello packets but also broadcasting packets. Analysis and computer simulation results show that the method using broadcast packets offers significant improvement over the method using Hello packet only. When the required hello packet transmission interval and the average broadcasting interval are equal, the overhead reduction is about 42%.

키워드

Wireless ad hoc network, neighbor discovery, Hello protocol, broadcasting

I. 서론

무선 에드혹 네트워크에서 동작하는 많은 프로토콜은 이웃 노드에 대한 정보를 중요하게 사용한다. 효율적인 flooding [1], GPS를 사용하는 라우팅 [2], 지향성 안테나를 사용하는 기법 [3] 등, 그 예는 무수히 많다.

이러한 이웃 노드에 대한 정보를 얻기 위하여, 일반적으로 무선 에드혹 네트워크의 각 노드는 Hello 패킷을 사용한다. 자신의 존재를 알리기 위하여 Hello 패킷을 전송하고, Hello 패킷을 수신한 노드는 메시지 내용을 통하여 이웃 노드에 대한 정보를 갱신한다. 하지만, 이러한 Hello 패킷의 전송은 주기적으로 이루어져야 하기 때문에 이는 네트워크에 큰 제어 부하로 작용하게 된다.

본 논문에서는 Hello 패킷뿐만 아니라 방송(Broadcasting) 패킷을 활용하여 이웃 노드 정보를 주고받는 방법을 고찰한다. 방송 패킷은 송신 노드의 모든 이웃 노드에게 전달되는데, 무선 에드혹 네트워크에서는 여러 가지 목적으로 방송 패킷이 사용된다. 예를 들면, 무선 채널을 선점하기 위하여 사용할 수 있는 RTS/CTS 패킷을 들 수 있다. 또한 무선 에드혹 네트워크에서 동작하는 많은 프로토콜에서는 Flooding 과정을 필요로 하는데, 이 Flooding을 수행하는 과정에서 방송 패킷이 사용된다.

분석과 컴퓨터 시뮬레이션을 통하여 방송 패킷을 활용하면 Hello 패킷을 전송하는 부하를 얼마나 감소시킬 수 있는지 고찰하였다. 만일 방송 패킷의 전송 주기가 Hello 패킷의 전송 주기와 같다면, 약 42% 정도의 부하 절감 효과가 있는 것을 알 수 있다.

본 논문의 구성은 다음과 같다. II 장에서는 배경 지식과 관련 연구에 대하여 기술하였다. III 장에서 Hello 패킷뿐만 아니라 방송 패킷을 활용하여 이웃 노드 정보를 주고받는 방법을 설명하고, IV 장에서 수학적 분석을, V 장에서 컴퓨터 시뮬레이션 결과를 제시하여 방송 패킷을 활용하는 것의 부하 절감 효과를 확인하였다. 마지막으로 결론을 맺고 있다.

II. 배경 지식

2.1 Hello 패킷

무선 에드혹 네트워크가 동작하기 위해서 필요로 하는 가장 기본적인 정보는 각 노드 간의 이웃 정보이다.

이러한 이웃 노드에 대한 정보는 이웃 노드가 전송하는 패킷을 성공적으로 수신함으로써 얻을 수 있는데, 이 때 사용되는 패킷을 Hello 패킷이라고 한다. 자신의 존재를 알리고 싶은 노드는 Hello 패킷을 주기적으로 전송한다. Hello 패킷을 수신한 노드는 이웃 노드의 존재를 감지하고, 자신의 이웃 노드 테이블을 갱신한다.

Hello 패킷을 전송하는 주기를 결정하는 것이 중요하다. Hello 패킷 전송 주기를 길게 하면, 이웃 노드에 대한 정보를 갱신하는데 걸리는 시간이 길어지게 된다. 따라서 노드의 이동이나 새로운 노드의 추가, 기존 노드의 비활성과 같이 네트워크 토폴로지 정보에 변화가 있을 때 이를 감지하는데 걸리는 시간이 길어지게 된다.

반대로 Hello 패킷의 전송 주기를 짧게 하면, 에드혹 네트워크 시스템의 부담이 증가하게 된다. Hello 패킷은 제어를 위한 패킷이므로, 실제 데이터를 주고받을 수 있는 링크 대역폭이 상대적으로 줄어들게 된다. 또한, 잦은 패킷의 전송은 송신하는 단말뿐만 아니라 수신측 단말의 전력 소모를 증가시킨다. 에드혹 네트워크는 단말의 배터리가 제한적이기 때문에 이는 에드혹 네트워크의 동작 기간(lifetime)을 단축시키게 된다. 따라서 에드혹 네트워크에서는 Hello 패킷 전송을 최대한 줄이면서도, 네트워크 토폴로지의 변화를 신속하게 감지하는 것이 이상적인 것을 알 수 있다.

2.2 관련연구

Chakeres 등은 [4]에서 무선 에드혹 네트워크의 각 노드가 서로 Hello 패킷을 주고받아 이웃 노드의 정보를 얻는 방법의 효율성에 대해서 연구하였다. 무선 에드혹 네트워크에서 Hello 프로토콜의 성능은 Hello 패킷의 크기, 전송 속도, 그리고 이웃 노드 테이블에 삭제하는 시간에 영향을 받는다고 주장하였다.

Giruka 등은 [5]에서 세 가지 새로운 Hello 프로토콜을 제시하였다. 첫째 일정 시간 간격으로 Hello 패킷을 전송하는 대신에 일정 거리마다 Hello 패킷을 전송하는 방식, 둘째 이웃 노드 정보를 항상 유지하는 대신에 필요한 경우에만 유지하는 방식, 셋째 일정시간 이내에 데이터 전송이 있었던 경우에만 Hello 패킷 전송을 계속하는 방식을 제시하였다. 그리고, 오버헤드와 정확성의 측면에서 각 프로토콜을 비교하였다. [6]에서도 앞의 첫째 방법과 유사하게 노드의 이동 속도에 따라 Hello 패킷의 전송 주기를 결정하는 방법을 제안하였다.

III. 프로토콜 동작

본 논문에서는 Hello 패킷뿐만 아니라 방송 패킷을 활용하여 이웃 노드 정보를 주고받는 방법을 제시한다. 무선 애드혹 네트워크에서는 여러 가지 목적으로 방송 패킷이 사용된다. 예를 들면, 무선 채널을 선점하기 위하여 사용되는 RTS/CTS 패킷을 들 수 있다. 또한 무선 애드혹 네트워크에서 동작하는 많은 프로토콜에서는 Flooding 과정을 필요로 하는데, 이 Flooding을 수행하는 과정에서 방송 패킷이 사용된다. Flooding은 무선 애드혹 네트워크 내의 모든 단말들에게 데이터를 전송하는 것을 목적으로 하기 때문에, 각 노드는 일반적으로 방송 패킷을 사용하여 자신의 모든 이웃 노드들에게 데이터를 전송하게 된다.

방송 패킷은 송신 노드의 모든 이웃 노드가 수신을 하기 때문에 송신 노드의 존재를 알리는 Hello 패킷의 역할을 대신하기에 충분하다. 따라서 방송 패킷을 전송한 경우에는 Hello 패킷 전송을 생략할 수 있다. 이는 제어 패킷인 Hello 패킷 전송을 줄여서, 무선 애드혹 네트워크의 부하를 줄이는 효과가 있다.

방송 패킷과 달리 일반 유니캐스트(Unicast) 패킷은 Hello 패킷의 역할을 대신하기에 적절하지 않다. 유니캐스트는 하나의 수신 노드를 지정하여 전송하는 방식이기 때문에, 지정된 수신 노드를 제외한 이웃 노드들은 수신된 패킷을 무시한다. 만일 노드들이 에너지 절약 모드로 동작하는 경우에는, 지정된 수신 노드 이외의 노드들은 자신에게 전송될 패킷이 없을 것으로 판단하고 에너지 절약을 위하여 무선 안테나를 통한 수신을 중단하고 슬립(Sleep) 상태로 있을 수도 있다. 따라서 유니캐스트 패킷은 모든 이웃 노드들에게 송신자의 존재를 알리는 것을 보장할 수 없다.

방송 패킷을 활용하여 Hello 메시지를 전달할 수 있는 프로토콜의 동작 과정은 알고리즘 1과 같다. 기본적으로 타이머를 사용하여 일정한 시간 간격 T_H 초마다 Hello 패킷을 전송한다. 만일 방송 패킷을 전송하게 되면 Hello 패킷을 전송한 것으로 간주하여 그 시점부터 T_H 초 후에 Hello 패킷을 전송하도록 타이머를 초기화한다. 패킷을 수신하는 노드는 패킷의 송신자 정보를 바탕으로 이웃 노드 테이블을 갱신한다. 이 동작은 패킷의 종류(Hello, 방송, 유니캐스트)와 무관하다.

알고리즘 1 노드의 동작 알고리즘

Algorithm 1. Node algorithm

```

1: function initialize()
2:     Hello 패킷 타이머 값을  $T_H$ 로 초기화
3: end function
    
```

```

1: function timeout()
2:     Hello 패킷 타이머 값을  $T_H$ 로 초기화
3:     Hello 패킷 전송
4: end function
    
```

```

1: function tx_broadcast_packet()
2:     Hello 패킷 타이머 값을  $T_H$ 로 초기화
3:     방송 패킷 전송
4: end function
    
```

```

1: function rx_packet()
2:     이웃 노드 테이블 갱신
3: end function
    
```

IV. 분석

앞서 방송 패킷을 활용하여 Hello 패킷의 전송을 줄이는 방법을 제시하였다. 이번 장에서는 수학적 분석을 통하여 Hello 패킷 전송 부하가 감소되는 정도를 고찰하고자 한다.

먼저 Hello 패킷만을 사용하여 이웃 노드 정보를 수집하는 경우를 고려하자. 이때에는 일정한 주기 T_H 초마다 한번씩 Hello 패킷을 전송한다. Hello 패킷이 전송되는 속도(rate)를 R_1 이라고 하면, 다음과 같이 간단히 계산할 수 있다.

$$R_1 = \frac{1}{T_H} \text{ (pkt/sec)} \tag{1}$$

다음으로 앞서 제안한 방송 패킷을 활용하여 Hello 패킷의 전송을 줄이는 방법을 고려한다. 한 노드가 방송 패킷을 전송하는 시간 간격을 랜덤 변수 X 라고 하자. 랜덤 변수 X 가 평균 T_B 인 지수 분포(exponential distribution)를 따른다고 하면, 다음의 관계가 성립한다.

$$P\{X \leq t\} = 1 - e^{-\frac{t}{T_B}} \tag{2}$$

어떤 방송 패킷을 전송한 순간부터 그 다음 방송 패킷을 전송하는 사이 구간에서 발생하는 Hello 패킷의 전송 회수를 고려하자. 제안한 알고리즘에 의하여 방송 패킷을 전송하면 그 뒤로 T_H 초 동안은 Hello 패킷이 전송되지 않는다. 그림 1은 본 논문의 분석 모델을 보여 주고 있다.

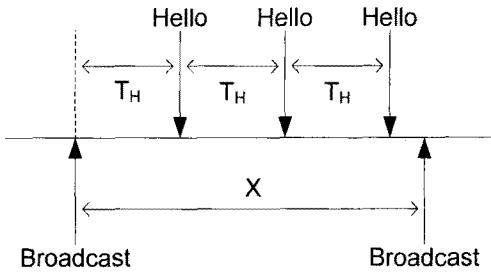


그림 1. 분석 모델
Fig. 1 Analysis model

어떤 방송 패킷을 전송한 순간부터 그 다음 방송 패킷을 전송하는 사이 구간에서 발생하는 Hello 패킷의 전송 회수를 랜덤 변수 H 라고 하면, 다음의 관계식이 성립한다.

$$P\{H=k\} = P\{kT_H < X \leq (k+1)T_H\} \quad (3)$$

단, k 는 0을 포함하는 양의 정수이다.

식(2)와 (3)에서 다음을 얻을 수 있다.

$$P\{H=k\} = e^{-k \frac{T_H}{T_B}} - e^{-(k+1) \frac{T_H}{T_B}} \quad (4)$$

방송 패킷 전송 사이에 전송하는 Hello 패킷의 평균 전송 회수는 다음과 같다.

$$E(H) = \sum_{k=0}^{\infty} kP(H=k) = \frac{e^{-\frac{T_H}{T_B}}}{1 - e^{-\frac{T_H}{T_B}}} \quad (5)$$

방송 패킷 전송 사이의 평균 간격이 T_B 이므로, 방송 패킷을 활용하여 이웃 노드 정보를 획득하는 경우에 Hello 패킷이 전송되는 속도를 R_2 라고 하면 다음과 같다.

$$R_2 = \frac{1}{T_B} \frac{e^{-\frac{T_H}{T_B}}}{1 - e^{-\frac{T_H}{T_B}}} \quad (pkt/sec) \quad (6)$$

Hello 패킷만을 사용하여 이웃 노드 정보를 수집하는 경우에 Hello 패킷이 전송되는 속도는 R_1 이고, 방송 패킷을 활용하여 이웃 노드 정보를 획득하는 경우에 Hello 패킷이 전송되는 속도는 R_2 이므로, 방송 패킷 전송을 활용하여 줄어든 Hello 패킷의 비율은 다음과 같다.

$$\frac{R_2}{R_1} = \frac{T_H}{T_B} \frac{e^{-\frac{T_H}{T_B}}}{1 - e^{-\frac{T_H}{T_B}}} = \frac{re^{-r}}{1 - e^{-r}} \quad (7)$$

단, $r = T_H/T_B$.

식 (7)에서 Hello 패킷 전송 부하 절감 효과의 크기는 Hello 패킷의 전송 주기와 방송 패킷의 평균 전송 주기의 비에 의하여 결정되는 것을 볼 수 있다.

IV. 성능 평가

앞서 방송 패킷을 활용하여 Hello 패킷의 전송을 줄이는 방법을 제시하고, 그에 대해 수학적 분석을 하였다. 이번 장에서는 컴퓨터 시뮬레이션을 통하여 Hello 패킷 전송 부하가 감소되는 정도를 고찰하고자 한다.

컴퓨터 시뮬레이션을 위하여 ns-2.31[7]을 사용하였다. DSDV 애드혹 라우팅 프로토콜[8]을 사용하는 무선 애드혹 네트워크를 가정하였다. DSDV 프로토콜은 실제 데이터 패킷의 전송이 없더라도 항상 모든 라우팅 테이블을 구성하여 유지하는(proactive) 방식으로 동작하는 애드혹 라우팅 프로토콜이다. 따라서, 링크 정보를 유지하기 위하여 주기적으로 Hello 패킷을 전송한다. 단, DSDV 프로토콜은 Hello 패킷을 전송할 때, 이웃 간의 동기화를 피하기 위하여 약간의 임의성(randomness)을 포함하여 전송 간격을 결정한다. 즉, 평균 시간 간격을 중심으로 약 14%의 편차를 허용하는 범위에서 임의의 시간을 선택하여 Hello 패킷의 전송 시간 간격을 결정한다. 분석에서는 분석의 편의를 위하여 정확한 간격을 유지하며 Hello 패킷이 전송된다고 가정하였는데, 실험 결과

를 통하여 이러한 가정의 영향을 관찰할 수 있다.

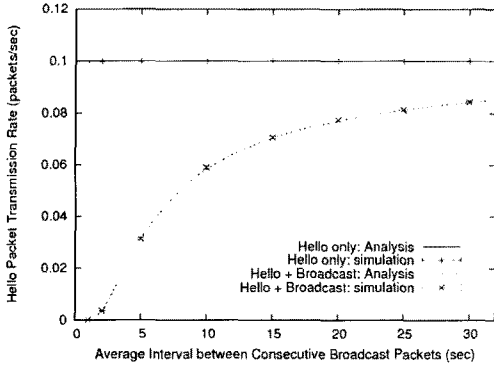


그림 2. 방송 패킷 전송 주기에 따른 성능 비교
Fig. 2 Performance comparison in terms of average broadcast packet transmission interval

먼저 Hello 패킷의 전송 주기(T_H)를 10초로 고정하고, 방송 패킷의 전송 주기(T_B)를 변화시키면서 Hello 패킷이 얼마나 자주 전송되는지를 살펴보았다. 그림 2에서 Hello 패킷만을 사용하는 기존의 DSDV 프로토콜과 Hello 패킷 뿐만 아니라 방송 패킷도 활용하는 알고리즘의 성능을 비교하였다. 본래의 DSDV 프로토콜은 방송 패킷의 전송과 무관하게 일정한 간격으로 Hello 패킷을 전송하기 때문에 Hello 패킷이 전송되는 속도가 0.1 (패킷/초)로 유지되는 것을 볼 수 있다. 반면에, 방송 패킷을 사용하여 Hello 패킷의 전송을 줄이는 방법은 방송 패킷의 전송 주기가 짧을수록 Hello 패킷의 전송이 급격하게 줄어드는 것을 볼 수 있다.

그림 2의 실험 결과는 동일한 환경에서 각기 다른 난수 초기값(random seed)을 사용하여 각 경우에 대하여 3번씩 시뮬레이션을 한 후, 실험 결과의 평균, 최소, 최대 값을 계산하여 나타내었다. 실험 간의 편차가 거의 없이 일관된 실험 결과 값을 생성하는 것을 볼 수 있다.

그리고, 앞서 시뮬레이션에서 DSDV 프로토콜이 Hello 패킷을 전송할 때, 14% 정도의 편차를 갖고 동작하는데 비해, 분석에서는 이를 일정한 값으로 가정하여 단순화하였다는 것을 언급하였다. 그림 2의 분석 결과와 시뮬레이션 결과를 비교해 보면, 분석의 편의를 위한 이 단순화 가정이 결과 값에 별다른 영향을 미치지 않는 것을 확인할 수 있다.

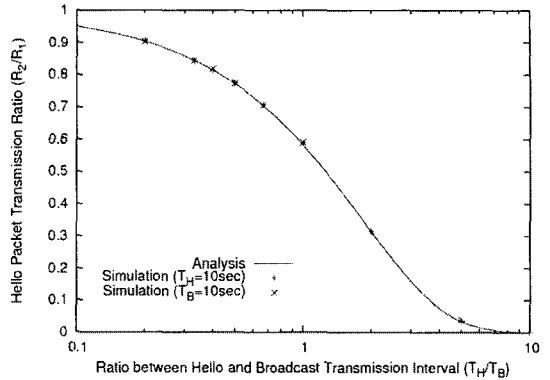


그림 3. Hello 패킷의 전송 주기와 방송 패킷 전송 주기의 비에 따른 부하 절감
Fig. 3 Overhead reduction in terms of the ratio between Hello and broadcast packet transmission interval

다음으로 식 (7)이 의미하는 바와 같이 Hello 패킷 전송의 부하 절감 효과의 크기가 Hello 패킷의 전송 주기와 방송 패킷의 평균 전송 주기의 비에 의하여 결정되는 것을 확인해 본다. 그림 3은 Hello 패킷의 전송 주기(T_H)를 10초로 고정하고 방송 패킷의 전송 주기(T_B)를 변화시킨 경우와, 반대로 방송 패킷의 전송 주기를 10초로 고정하고 Hello 패킷의 전송 주기를 변화시킨 경우의 시뮬레이션 결과를 분석 결과와 비교하였다. Hello 패킷의 전송 주기를 고정시킨 경우나 방송 패킷의 전송 주기를 고정시킨 경우에 상관없이 두 주기 사이의 비율에 의하여 Hello 패킷을 전송하는 부하가 절감되는 비율이 결정되는 것을 확인할 수 있다. 예를 들어 Hello 패킷의 전송 주기와 방송 패킷의 전송 주기가 같다면, Hello 패킷을 전송하여야 하는 빈도가 약 42% 정도 감소한다. 방송 패킷의 전송 주기가 Hello 패킷 전송 주기의 두 배 정도로 큰 값이더라도, Hello 패킷을 전송하여야 하는 빈도를 약 23% 정도 줄일 수 있게 된다.

V. 결 론

본 논문에서는 방송 패킷을 활용하여 무선 애드혹 네트워크에서 이웃 노드 정보를 유지하기 위하여 주기적으로 전송하는 Hello 패킷을 줄이는 방안을 고찰하였다. 방송 패킷을 활용하는 방법에 대하여 수학적 분석과 컴퓨터 시뮬레이션을 통하여 Hello 패킷을 전송하는 부

하를 절감하는 효과가 큰 것을 보였다. 만일 방송 패킷의 전송 주기가 Hello 패킷의 전송 주기와 같다면, 약 42% 정도의 부하 절감 효과가 있는 것을 알 수 있었다.

참고문헌

- [1] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," Proc. of ACM MSWiM '00, pp. 61 - 68, 2000.
- [2] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proc. of MOBICOM '00, pp. 243 - 254, 2000.
- [3] R. Choudhury, X. Yang, R. Ramanathan, and N. Vadiya, "Using Directional Antennas for Medium Access Control in Ad Hoc Networks," Proc. of MOBICOM '02, pp. 59 - 70, 2002.
- [4] I. Chakeres and E. Belding-Royer, "The Utility of Hello Messages for Determining Link Connectivity," Proc. of WPMC '02, pp. 504 - 508, 2002.
- [5] V. Giruka and M. Singhal, "Hello Protocols for Ad-Hoc Networks Overhead and Accuracy tradeoffs," Proc. of WoWMoM '05, pp. 354 - 361, 2005.
- [6] F. Ingelrest and N. Mitton, and D. Simplot-Ryl, "A Turnover based Adaptive HELLO Protocol for Mobile Ad Hoc and Sensor Networks," Proc. of Mascots '07, 2007.
- [7] The network simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.
- [8] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Proc. of ACM SIGCOMM '94, pp. 234 - 244, 1994.

저자소개



최선웅(Sun-Woong Choi)

2005년 서울대학교 전기, 컴퓨터 공학 (공학박사)

2005년 9월 ~ 2007년 2월 삼성전자 정보통신총괄 책임연구원

2007년 3월 ~ 현재 국민대학교 전자공학부 전임강사
※ 관심분야: 무선 네트워크, 네트워크 자원관리, 시스템 성능 평가