

구성노드의 부하균등을 고려한 Ad-hoc 라우팅 프로토콜 제안

김인수⁰ 김시관 오길호
금오공과대학교 컴퓨터공학부

iskim⁰@cespc1.kumoh.ac.kr, sgkim@se.kumoh.ac.kr, gilho@cespc1.kumoh.ac.kr

A Proposal of Ad-hoc Routing Protocol with Load Balancing of Constituent Nodes

In-Soo Kim⁰ Si-Gwan Kim Gil-Ho Oh

School of Computer Engineering, Kumoh National Institute of Technology

요 약

Ad-hoc 네트워크에서 노드간의 효율적인 통신을 제공하기 위한 라우팅 프로토콜에 관한 연구가 활발하게 진행되고 있다. 현재 사용되고 있는 Ad-hoc 네트워크에서의 라우팅 프로토콜은 토폴로지가 변화해야만 새로운 경로를 설정하도록 되어 있어 노드 이동성이 감소하면 패킷 전송률은 증가하고 라우팅 오버헤드는 줄어들게 된다. 하지만, 일부 소수 노드들에게 트래픽이 집중되는 문제가 발생하기 때문에 패킷 전송 지연은 오히려 증가하게 된다. 본 논문에서는 각 노드들이 부하균등을 고려하여 혼잡 상태를 피해 경로를 설정하는 방법과 설정된 경로를 통해 정보가 전달되는 과정에서 경로단절 문제가 발생할 때 빠르게 대체 경로를 통해 전달되는 방법을 제안한다.

1. 서 론

무선 Ad-hoc 네트워크는 중앙 집중화된 관리자 표준화된 지원 서비스를 사용하지 않고 임시 망을 구성하는 무선 이동 호스트들의 집합을 말한다. 따라서 무선 Ad-hoc 네트워크는 기존의 유선 네트워크의 도움 없이 필요할 때 빠르게 구성될 수 있는 네트워크이다. 유선 네트워크를 구성하기 어려운 군사 통신이나 단기간 사용되는 사업장, 교육 통신 등을 위해 무선 Ad-hoc 네트워크가 사용된다.

Ad-hoc 네트워크는 백본 호스트나 다른 이동 호스트로의 연결을 제공하기 위한 고정된 제어 장치를 갖지 않으며, 각 이동 호스트가 라우터로 동작하여 이동 호스트로부터의 패킷을 다른 이동 호스트로 전달한다. 한 개 이상의 경로를 형성하는 이동 호스트가 다른 곳으로 이동함으로써 해당 경로를 무효화시키기 때문에 이러한 네트워크에서의 통신 연결은 상당히 취약하다. 이동 호스트는 이동에 따른 경로의 계산과 수정에 많은 시간을 소비하게 되면 데이터 처리율이 낮아져서 비효율적이고 비현실적인 시스템이 되기 쉽다. 따라서 Ad-hoc 이동 호스트들간에 높은 효율의 통신을 제공하기 위해서 잘 정의된 라우팅 기법이 필요하다.

Ad-hoc 네트워크를 위한 라우팅 알고리즘에는 순향적(proactive) 방식과 반응적(reactive) 방식, 두 설정 방식을 혼합한(hybrid) 방식으로 분류할 수 있다.

순향적 방식은 테이블 기반(Table-driven) 라우팅 방법으로 각 이동 호스트가 자신의 라우팅 정보를 주기적으로 브로드캐스팅하여 각 이동 호스트로 하여금 네트워크 내의 경로 정보를 항상 유지해서 필요할 때 즉시 경로 정보를 사용할 수 있는 방법이다. 이러한 방식은 경로에 대한 최신 정보를 항상 유지하기 위해 많은 네트워크 트래픽을 발생시키며 라우팅 정보의 방송 주기가 길어지면 대부분의 경로 정보가 쓸모 없게 되는 문제점을 갖는다. 대표적인 순향적 방식의 라우팅 프로토콜에는

DSRV(Destination Sequence Distance Vector)가 있다[1].

반응적 방식은 요청 기반(On-demand) 라우팅 방법으로 임의의 시점에 한 노드가 다른 노드로 데이터 전송을 원할 때만 경로를 설정한다. 모든 경로를 항상 유지하지 않기 때문에 트래픽을 감소시키며 상대적으로 적은 메모리 공간을 필요로 한다. 하지만 경로 설정 과정에서 경로 설정 지연 시간이 발생하며, 데이터 전송 중 발생하는 경로 실패로 인해 경로 재설정 작업이 요구될 수 있다. 반응적 방식의 라우팅 프로토콜에는 DSR(Dynamic Source Routing), AODV(Ad-hoc On-Demand Distance Vector Routing), TORA(Temporally Ordered Routing Algorithm) 등이 있다[2][3][4].

대표적인 라우팅 방식은 On-demand 방식이며 그 중에서도 DSR과 AODV가 가장 두드러진다. 이 프로토콜들은 노드들의 이동성이 낮을수록 패킷 전송률은 증가하고 반대로 라우팅 오버헤드는 감소한다. 그러나 노드의 이동성이 낮아져도 패킷 전송 지연이 오히려 증가하는 문제점이 있다[5]. 이는 부하 균등(load-balancing)을 고려하지 않았으므로 트래픽이 특정 노드들에게 집중되기 때문에 생기는 현상이다. 트래픽의 집중 현상은 전송 지연을 발생시키고 특정 노드들에게 많은 작업 오버헤드와 배터리 소모를 가져오게 한다.

본 논문에서는 Ad-hoc 네트워크 환경에서 각 노드들이 부하 균등을 고려해서 경로를 설정하므로 트래픽이 집중되는 경로를 피할 수 있고, 링크 단절 문제도 빠르게 우회경로를 선택해서 전달하는 방법을 제안한다.

2. 관련 연구

기존의 DSR이나 AODV와 같은 라우팅 프로토콜들은 노드의 이동성이 높은 환경에서 On-demand 방식을 통하여 네트워크 토폴로지가 변경될 때마다 새로운 경로 설정을 위한 경로 탐색 매커니즘을 수행하여 지연시간이 가장 작은 경로를 새롭게 선택

함으로써 노드에 미치는 부하 문제를 해결할 수 있었다. 하지만 노드의 이동성이 낮은 안정된 상황에서는 특정 노드에 트래픽이 집중되어 혼잡상태에 빠져도 이를 해결하기 위한 새로운 경로를 설정하는 매커니즘이 포함되어 있지 않기 때문에 노드의 혼잡 상태는 이동성이 낮을수록 점점 악화되며 이로 인하여 전체 네트워크 성능 저하를 초래한다.

LBAR(Load-Balanced Wireless Ad Hoc Routing)은 부하균등을 고려한 대표적인 라우팅 프로토콜이다[6]. LBAR은 경로를 설정하기 위해 각 노드의 활동성(node activity)과 트래픽 간섭(traffic interference)의 비율을 산출해서 트래픽 로드가 가장 작은 경로를 찾아내는 것을 제안하고 있다. 노드 활동성은 해당 노드의 현재 트래픽 로드 상태를 나타내며, 트래픽 간섭은 현재 노드에 이웃하고 있는 이웃 노드들의 트래픽 로드의 합을 나타낸다. LBAR은 경로 설정의 기준으로 자신과 이웃 노드들의 트래픽 로드 정보를 수집하기 위한 부가적인 오버헤드가 발생한다. 또한 목적지 노드의 라우팅 테이블에는 하나 이상의 경로 정보를 유지하기 위한 오버헤드와 링크 단절이 발생할 때 노드의 이동으로 네트워크 토폴로지가 바뀔 때 오류 발생 가능성이 높아진다.

3. 부하균등을 고려한 라우팅 프로토콜

부하균등을 고려한 라우팅 프로토콜은 경로를 설정하기 위해 각 노드의 트래픽을 고려하는 부분, 각 노드에서 라우팅 테이블을 유지하는 방법 그리고 링크 단절이 발생할 때 대처하는 방법으로 구성되어 있다.

소스는 이웃 노드들에게 hello message를 전송하고, hello message를 받은 노드들은 다시 이웃 노드들에게 브로드캐스팅한다. 목적지에서는 받은 메시지들을 라우팅 테이블에 저장하고 트래픽 복잡도가 가장 작은 경로를 선택해서 ACK 메시지를 소스에 전송한다. ACK 메시지를 받은 소스는 경로가 설정되었음을 알고 데이터 패킷을 전송한다.

3.1 소스 노드 알고리즘

소스 노드는 경로가 설정되어 있지 않은 목적지와 통신할 때 우선 경로 탐색 매커니즘을 수행한다. 소스 노드는 홉 카운트 값을 고려하여 hello message를 전송하게 된다. 소스에서 목적지까지 경로를 찾고 데이터를 전송하는 알고리즘은 다음과 같다.

- ① 소스는 자신의 이웃 노드들에게 hello message를 브로드캐스팅한다.
- ② hello message를 수신한 노드는 홉 카운트 값을 1 증가시키고 이웃하는 노드들에게 다시 hello message를 전송한다.
- ③ 각 중계 노드는 라우팅 테이블에 노드의 리스트를 포함한 각종 정보를 저장하고, 다시 이웃 노드들에게 hello message를 브로드캐스팅한다.
- ④ 중계 노드는 홉 카운트 값을 비교하여 자신의 홉 카운트 값보다 작은 값을 가지고 전송되는 노드의 정보만 수신한다.
- ⑤ 만약 같은 홉 카운트 값을 가지고 있는 노드들이 동시에 메시지를 전송하면 라우팅 테이블에 각각의 정보를 저장한 후 트래픽 복잡도가 작은 값을 가지는 것을 다시 이웃 노드들에게 포워딩시킨다.
- ⑥ 소스 노드는 목적지로부터 ACK를 수신하면 설정된 경로를 따라 데이터 패킷 전송을 시작한다.

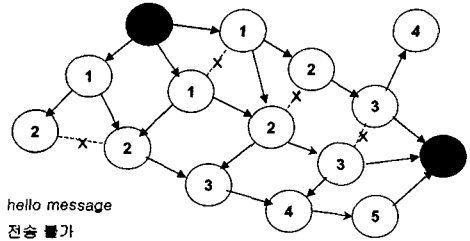


그림 1. 경로 탐색 과정

그림 1은 소스에서 목적지까지 경로 탐색 과정을 보여주고 있다. 기존의 부하균등 라우팅 프로토콜(LBAR)에서는 방문하지 않은 모든 노드에 setup message를 전송하므로 많은 경로를 가질 수 있는 반면에 목적지에서 각 정보를 라우팅에 저장하는 시간과 경로 유지비용이 많이 드는 문제가 있었다. 제안하는 방법에서는 각 노드마다 홉 카운트 값을 주어서 목적지로 가는 중간에도 최적의 경로를 구성하면서 경로를 탐색하는 방법을 사용하고 있다.

```

if (S has new packets to send) {
    while (route discovery waiting period) {
        forward hello message to all neighbor nodes;
        /* neighbor nodes = N*/
        if (receives hello message) {
            HC +1; /* hop count = HC */
            if (current node HC > previous node HC)
                receives hello message and forward;
            else
                discard hello message;
        }
    }
}
if (S receives ACK from D)
    start data transmission;
if (S does not receive ACK from D)
    restart route discovery;
    
```

알고리즘 1. 소스 노드의 경로 탐색 과정

3.2 목적지 노드 알고리즘

목적지 노드에 도착한 각 노드들의 정보는 라우팅 테이블에 저장하고 가장 작은 트래픽 복잡도를 갖는 경로를 선택해서 소스에 ACK를 전송한다. 전송 과정에서 네트워크 토폴로지 변화와 같은 링크 단절 문제가 발생하면 중계 노드에 있는 라우팅 테이블에서 다시 경로를 선택해서 전송한다. 목적지에서 소스로 전송하는 알고리즘은 다음과 같다.

- ① 목적지 노드는 라우팅 테이블에 각각의 hello message를 저장한다.
- ② 라우팅 테이블을 기반으로 경로 선택 시간 내에 최적의 경로를 선택하고 ACK 메시지를 소스로 전송한다.
- ③ 전송 과정에서 중계 노드사이에서 문제가 발생하면 목적지로 여러 메시지(ERR)를 전송한다.
- ④ ERR 메시지를 받은 목적지 노드는 라우팅 테이블에서 여러가 난 노드를 포함하고 있는 경로 테이블을 삭제한다.

⑤ 목적지로 ERR 메시지를 전송하는 동안에 링크 문제를 탐지한 중계 노드는 자신이 관리하는 라우팅 테이블에서 경로를 설정해서 ACK 메시지를 전송한다.

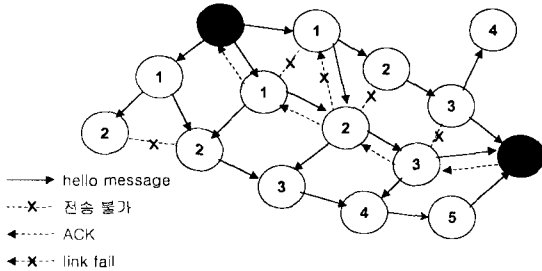


그림 2. 링크 단절시 우회경로 선택 과정

그림 2는 목적지 노드에서 스스로 ACK를 전송하는 과정을 보여주고 있다. 전송 과정 중에 중계 노드에서 링크 단절 문제가 발생하면 기존의 방법에서는 목적지에서 에러가 난 링크를 포함하는 경로 테이블을 삭제하고 두 번째로 비용함수가 적은 경로를 선택해서 ACK를 전송을 한다. 본 논문에서 제안한 방법은 각 중계 노드가 라우팅 테이블을 유지하고 있기 때문에 바로 경로를 선택해서 ACK를 전송한다.

```

if (D receives hello message)
  store at the route table;
  select path with best complexity;
  send ACK to source;
if (D receives ERR message) {
  remove paths containing broken links at the route tables;
  if (D has alternate path)
    select second complexity path and send;
  else
    send ERR message to source;
}
    
```

알고리즘 2. 목적지 노드의 ACK 메시지 전송

```

/* intermediate node = I */
if (node I receives hello message) {
  if ( I receive two hello message) {
    if(current node HC > previous node HC) {
      receives and forward to N;
    }
    else
      discard;
  }
  else
    forward to N;
}
}
if (node I receives ERR message)
  backward to D;
select best complexity path and send to S;
    
```

알고리즘 3. 중계 노드의 전송 과정

3.3 트래픽 복잡도 함수

트래픽 복잡도 함수는 노드의 이동성이 적은 Ad-hoc 네트워크에서 특정 노드로의 트래픽 집중화 현상을 피하기 위해 가능한 한 트래픽 부하가 가장 낮은 경로를 선택하기 위해 사용된다. 트래픽 복잡도 함수는 다음과 같이 정의한다.

- 노드 홉 카운트 H_i : 노드 i 의 순서 번호
- 노드 활동성 A_i : 소스에서 목적지로 가는 경로 상에 있는 노드의 수
- 트래픽 간섭 TI_i : 노드 i 에 이웃하고 있는 이웃 노드들의 트래픽 로드의 합, j 는 노드 i 의 이웃하는 노드들

$$TI_i = \sum_j A_j$$

- 트래픽 복잡도 C_k : 경로 k 의 트래픽 복잡도

$$C_k = \sum_{i \in k} (H_i + A_i + TI_i) = \sum_{i \in k} (H_i + A_i + \sum_j A_j)$$

4. 결론 및 향후 과제

Ad-hoc 네트워크에서 기존의 라우팅 프로토콜들은 현재 설정된 경로가 혼잡하다 하더라도 네트워크 토폴로지가 변하기 이전에 새로운 경로를 설정하지 않는다. 노드의 이동성이 감소하면 패킷 전송률은 증가하고 라우팅 오버헤드는 줄어들게 되지만, 특정 노드에 트래픽이 집중되는 문제가 발생하기 때문에 패킷 전송 지연은 오히려 증가하게 된다. Ad-hoc과 같은 무선 네트워크에서는 CPU 성능과 배터리 수명 등 한정된 자원을 사용하기 때문에 치명적인 문제가 될 수 있다.

본 논문에서는 트래픽 복잡도를 이용하여 혼잡 상태에 있는 경로를 우회해서 트래픽 집중 현상을 완화시키는 기존의 부하 균등 알고리즘을 홉 카운터와 각 중계 노드가 라우팅 테이블을 관리하는 방법을 제시하고 있다. 각 노드가 홉 카운터를 이용하여 hello message를 전송함으로써 브로드캐스팅 오버헤드를 줄일 수 있어 효율적인 대역폭을 사용할 수 있고, 중계 노드에서 각각의 라우팅 테이블을 관리함으로써 경로 단절 문제가 발생할 때 빠르게 대체경로를 설정할 수 있는 방법을 제안하였다. 현재 본 논문에서 제안한 방법에 대한 수치적 실험 결과 분석과 이동성이 높은 환경에서 효과적인 구성노드의 부하균등 기법에 대한 연구가 진행 중에 있다.

5. 참고 문헌

- [1] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", ACM SIGCOMM, vol. 24, no. 4, Oct. 1994.
- [2] J. Broch, D. B. Johnson and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet Draft, Dec. 1999.
- [3] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc On Demand Distance Vector(AODV) Routing", Internet Draft, Nov. 1999.
- [4] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm(TORA) Functional Specification", Internet Draft, Aug. 1998.
- [5] IETF MANET Working group Charter <http://www.ietf.org/html.charters/manet-charter.html>
- [6] Audrey Zhou and Hossam Hassanein, "Load-Balanced Wireless Ad Hoc Routing", Proc.Canadian Conference on Electrical and Computer Engineering, vol. 2, pp1157-1161, 2001.