

무선 애드 hoc 네트워크에서의 다중전송속도 플러딩 알고리즘

남용섭, 최낙중^o, 권태경, 최양희

서울대학교 전기컴퓨터공학부

Multi-Rate Flooding Algorithm for Mobile Ad Hoc Networks

Yongsub Nam, Nakjung Choi^o, Taekyoung Kwon and Yanghee Choi

School of Computer Science and Engineering, Seoul National University

{ysnam, , fomula, tk, yhchoi}@mmlab.snu.ac.kr

요 약

In mobile ad hoc networks (MANETs), due to their characteristics of mobility, a number of messages are transmitted in a flooding manner. Although IEEE 802.11 Wireless LANs provide multiple transmission rates depending on the channel conditions, broadcast frames are transmitted in a rate selected among the basic rate set. Therefore, broadcast frames can bring about high latency of broadcast messages as well as severely deteriorate the aggregate throughput of the network. We propose a multi-rate flooding algorithm for MANETs in order to enhance the system throughput and delay of MANETs, by choosing the optimal data rate according to the neighbor channel conditions. In addition, we reduce the redundant broadcast messages by attaching the neighbor list in the broadcast message. Analysis and simulation results confirm that the proposed algorithm achieves significantly less delay than basic-rate flooding algorithm.

1. 개요

이동 멀티홉 애드 hoc 네트워크(MANET)는 무선 단말들이 이동하는 특성을 가지고 있으므로 시간에 따라 네트워크 토폴로지가 변화하는 특징을 갖는다. 따라서 MANET 의 경우, 플러딩을 통한 메시지 전송의 비중이 다른 네트워크에 비해 상대적으로 높다. MANET 을 위해 제안된 라우팅 프로토콜인 AODV, DSDV 등에서도 라우팅 경로 탐색 및 관리를 위해 플러딩을 통한 패킷 전송이 빈번히 발생한다.

MANET 에서 제 2 계층 프로토콜로 널리 사용되는 IEEE 802.11 표준 [1] 은 브로드캐스트 혹은 멀티캐스트의 경우, 명시적인 MAC ACK 메시지 전송을 정의하지 않으며, 가급적 모든 단말들이 메시지를 들어야 하기 때문에 기본 전송 속도 집합(basic rate set)에서 전송 속도를 선택한다. 따라서, MANET 브로드캐스트의 경우, 일반적으로 낮은 전송 속도를 선택하기 때문에 메시지 전송에 걸리는 시간이 길어질 뿐만 아니라 이로 인해 채널을 점유하는 시간 역시 길어지기 때문에 전체 시스템 성능도 저하된다.

IEEE 802.11 a/b/g 표준 [1]은 물리 계층에서 코딩 방식에 따라 여러 단계의 전송 속도를 제공한다. 예를 들어, IEEE 802.11b 표준의 경우, 1, 2, 5.5, 11Mbps 의 4 가지 전송 속도를 정의하고 있으며, 각 벤더들이 정의하는 무선 채널 적응 기법(ex. ARF:

Auto-Rate Fallback)을 사용하여 전송 속도를 결정한다. 현재까지 다중 전송 속도를 활용하는 연구가 이루어져 왔으나, MANET 브로드캐스트를 위한 방법은 제안된 적이 없다.

본 연구는 IEEE 802.11 의 다중 전송 속도 특성을 활용하여 MANET 에서 다중 전송 속도 플러딩을 수행하는 알고리즘을 제안한다. 이를 위해 각각의 무선 단말이 이웃 단말들과의 채널 상태를 저장 및 관리하고, 이를 바탕으로 최적의 전송 속도를 계산한다.

본 논문은 다음과 같이 구성된다. 2 장에서 관련 연구를 살펴보고 3 장에서 제안한 알고리즘을 소개한다. 4 장에서 분석과 모의 실험을 통해 제안 알고리즘의 성능을 평가하고, 마지막으로 5 장에서 본 논문을 마무리 짓는다.

2. 관련 연구

무선 네트워크에서의 다중 전송 속도 활용을 위해 제안된 대표적인 기법으로 [2] [3]이 있다. MAS [2]는 MANET 환경에서 유니캐스트 패킷의 전송 시 다중 전송 속도 특성을 사용하여 더 빠른 경로를 이용하는 방법으로 제안된 알고리즘이다. MAS 는 네트워크 계층의 하위에 위치하며, 라우팅 알고리즘에서 결정된 다음 홉을 변경하여 더 빠른 경로로 릴레이하며, 릴레이 단말은 해당 패킷을 다시 원래의 다음 홉으로 전송한다. 이러한 방식으로 다중 전

송 속도를 활용하여 전송 시간의 이득을 얻는다.

rDCF [3]는 MAC 계층에서 다중 전송 속도를 활용하는 알고리즘이다. rDCF의 기본 알고리즘은 MAS와 동일하나, MAS가 네트워크 계층 하단에 위치하는데 반해 rDCF는 MAC 계층에서 릴레이를 수행한다. 즉, MAC 계층에서 다음 홉으로 프레임 전달 시 더 빠른 전송 속도를 갖는 릴레이 단말을 통해 2 홉으로 전송한다.

플러딩 기법으로 제안된 대표적인 기법으로 [4] [5] [6] [7] [8] [9] [10] [11] [12]가 있다. [4] [5]는 가장 기본적인 플러딩 기법으로 이미 이전에 받은 브로드캐스트 메시지를 다시 전송하지 않는 방법을 사용한다. 이들은 전송 단말의 주소와 시퀀스 번호만을 기록하는 방법을 사용하여 구현이 단순하나, 모든 단말이 최소 한번은 전송을 수행하여 과도한 트래픽을 유발하는 단점이 있다.

[6] [7] [8]은 확률에 기반하여 브로드캐스트 횟수를 줄이는 플러딩 기법을 제안한다. 이들 기법들은 브로드캐스트 횟수를 효과적으로 줄일 수 있으나 네트워크에 맞는 파라미터 설정이 어려운 문제이며, 또한 확률에 기반하므로 경우에 따라 모든 단말에게 메시지가 전송됨을 보장할 수 없다.

[9]는 1 홉 이웃 단말들에 대한 정보를 바탕으로 전송 범위의 주변에서만 브로드캐스트를 수행하는 기법을 제안하고 있다. 브로드캐스트 횟수를 효과적으로 줄일 수 있으나, 각 단말에 대한 위치와 거리 정보를 알아야 하는 단점이 있다.

[10]은 [9]와 마찬가지로 1 홉 이웃 단말에 대한 정보를 바탕으로 하여 플러딩을 수행하나 [9]와 달리 거리 정보를 필요로 하지 않는다. 대신 브로드캐스트를 수행할 때 자신의 1 홉 이웃들의 목록을 메시지에 첨부하여 전송한다. 이 메시지를 받은 단말은 자신의 1 홉 이웃들의 목록과 메시지의 목록을 비교하여 플러딩 수행 여부를 결정한다. 브로드캐스트 횟수를 최적으로 줄이지 못하며 메시지 헤더의 오버헤드가 발생하나 이동성이 있는 MANET에 있어 적합한 방법으로 고려할 만하다.

[11] [12]는 2 홉 이상의 이웃 단말에 대한 정보를 관리하며 이를 바탕으로 플러딩을 수행한다. 최적에 가깝게 브로드캐스트 횟수를 줄일 수 있으나 단말의 이동성이 높은 경우 바람직한 접근 방법이 아니며, 또한 단말 목록과 연산의 오버헤드가 상당히 큰 단점이 있다.

3. 제안 알고리즘

기본적인 알고리즘은 플러딩을 하고자 하는 단말이 1 홉 내 이웃들 중 2 홉으로 전송하는 경우가 더 빠른 경우 해당 이웃을 제외하고 브로드캐스트하는 방식으로 동작한다. 즉, 단말 자신이 1 홉 내 이웃들에게 브로드캐스트 하고 해당 이웃이 다시 목적 단말에게 전송하는 경우가 더 빠른 경우 해당

경로를 택하는 방식이다. 각 단말은 자신의 1 홉 이웃들의 목록 중에서 직접 1 홉으로 보내는 것보다 다른 이웃을 거쳐갈 경우 더 빨리 보낼 수 있는 단말들을 모두 제거한 상태에서 필요한 가장 낮은 전송 속도를 택해 브로드캐스트를 수행한다.

이를 위해 필요한 이웃 단말 테이블 관리와 전송 속도 계산 알고리즘은 다음 장에 소개한다.

3.1 Link 정보 관리

각각의 단말들은 주기적인 Neighbor discovery 메시지 교환을 바탕으로 자신의 2 홉 이웃들의 전송 속도를 저장하고 관리한다. 2 홉 내 Neighbor discovery는 다음과 같은 과정을 통해 수행된다. 우선, 각각의 단말들이 주기적으로 Neighbor discovery 메시지를 브로드캐스트 한다. 이를 받은 1 홉 내 단말들은 자신과 전송 단말 사이의 전송 속도를 Neighbor Reply 메시지에 담아 응답한다. 이러한 방법으로 각각의 단말들은 1 홉 내 이웃들과의 전송 속도 테이블을 저장한다. (두 단말 사이의 전송은 대칭이라고 가정한다.) 이와 별도로 각 단말들은 주기적으로 자신의 1 홉 이웃들과의 전송 속도 테이블을 담은 Neighbor notify 메시지를 1 홉 내 이웃들에게 브로드캐스트 한다.

Neighbor notify 메시지를 받은 각 단말들은 neighbor 테이블 내의 각 엔트리에 일정 유효기간을 부여하여 자신의 테이블에 추가한다. (엔트리의 유효기간은 notify 메시지 전송 주기의 2 배로 설정한다.) 이때, neighbor table의 각 엔트리는 {단말의 pair, 남은 유효기간, 전송속도}로 이루어진다. 유효기간이 지나가는 동안 업데이트 메시지를 받지 못한 경우 해당 엔트리는 테이블에서 제거된다. 이러한 방법으로 각 단말들은 자신의 2 홉 내 이웃 단말들에 대한 전송 속도를 저장 및 관리하게 된다.

3.2 전송 속도 결정 알고리즘

전송 속도 결정 시, 각 단말은 in-set (Q_{in})과 out-set (Q_{out})의 두 개의 리스트를 활용한다. 우선, 플러딩을 수행하고자 하는 단말은 자신의 1 홉 내 이웃들의 목록을 Q_{in} 에 저장한다. 초기에 Q_{out} 은 비어있는 상태로 시작된다. 해당 단말은 Q_{in} 내의 단말 중 자신과 가장 전송 속도가 낮은 단말을 선택한 후, 해당 전송 속도를 기록한 뒤, 이 단말을 Q_{in} 에서 삭제하여 Q_{out} 에 추가한다. 그리고 Q_{out} 내의 모든 단말이 Q_{in} 내의 임의의 단말을 통해 더 빠른 시간에 전송 가능한지 확인한다. 만약 더 빠른 경로가 발견될 경우, 앞의 과정을 반복한다. 즉, 다시 가장 낮은 전송 속도를 갖는 단말을 선택하여 과정을 수행한다. 만약 더 빠른 경로가 없다면, 해당 시점에서 저장하고 있던 전송 속도를 최종 전송 속도로 결정한다.

```

// determines the data transmission rate
DETERMINE_RATE (NT, LT)
  Qin ← NT
  Qout ← ∅
  Rm ← MAX_RATE
  while Qin ≠ ∅
    Rm ← MAX_RATE
    // extract minimum-rate one-hop node
    for each node v ∈ Qin
      Rc ← data_rate between (this node, v)
      if Rc < Rm
        then Rm ← Rc, u ← v
    Qin ← Qin - {u}
    Qout ← Qout ∪ {u}
    // check connectivity
    for each node v1 ∈ Qout
      conn ← false
      for each node v2 ∈ Qin
        if one-hop path exists between (v1, v2) in LT
          & 1/Rate(this node, v2) + 1/Rate(v2, v1) < 1/Rate(this node, v1)
            then conn ← true; break
      if conn ≠ true
        then return Rm
  return Rm

```

그림 1. 다중전송속도 플러딩 알고리즘

그림 1 은 위에서 언급한 전송 속도 계산 알고리즘을 기술한 것이다. 여기에서, NT, LT는 각각 1 홉 이웃들의 목록과 자신이 저장하고 있는 2 홉 내 모든 링크에 대한 목록을 나타낸다. Q_{in}과 Q_{out}은 전송 속도 계산 시 사용되는 이웃들에 대한 목록 중 현재 그룹에 포함된 단말과 포함되지 않은 단말의 목록을 의미한다. 마지막으로 MAX_RATE는 MAC 프로토콜에서 지원하는 가장 높은 전송 속도를 의미한다.

3.3 플러딩 알고리즘

플러딩 방법은 [10]에서 제안한 self-pruning 방법을 응용한다. 즉, 브로드캐스트 메시지의 헤더에 최초 전송 단말의 주소와 시퀀스 넘버, 그리고 보내는 단말의 1 홉 이웃들에 대한 목록을 첨부하여 전송한다. 이를 받은 단말은 우선 최초 전송 단말의 주소와 시퀀스 넘버를 확인하여 자신이 이전에 받은 메시지인지 확인한다. 만약 이전에 받은 적이 없는 메시지인 경우, 메시지의 단말 목록과 자신의 1 홉 이웃 목록을 비교하여 메시지 브로드캐스트 여부를 결정한다.

브로드캐스트가 결정된 경우, 임의의 확정 기간을 두어 메시지의 충돌을 막으며, 이 기간 동안에 다른 단말의 브로드캐스트 메시지를 엿들어 전송 포기 여부를 결정한다. 즉, 자신의 1 홉 이웃들 중 브로드캐스트 메시지의 목록에 포함되지 않는 단말들이 해당 목록에 포함되어 있는지 확인하여 만약 해당 브로드캐스트 메시지의 수신 영역에 포함되어 있으면 메시지 전송을 포기한다.

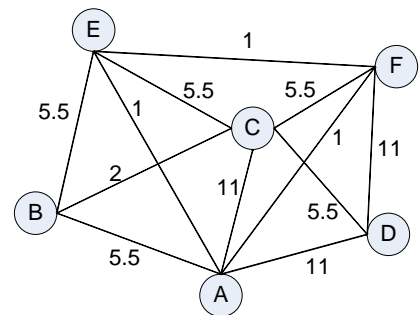
이웃 단말의 목록 전송으로 인한 오버헤드가 야기되거나 이동성이 있는 네트워크의 경우 효과적으

로 플러딩이 가능하다.

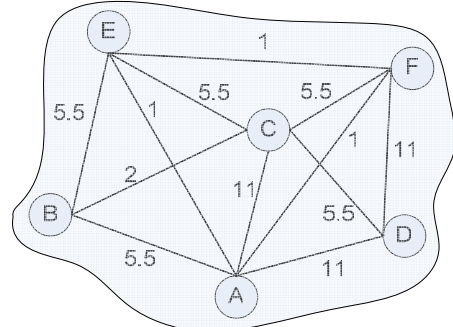
3.4 제안 알고리즘의 동작 예

그림 2 는 제안한 알고리즘의 동작 예를 보인다. 그림 2 의 (a)와 같이 네트워크 토폴로지가 구성되어 있으며 단말 A 가 플러딩을 수행한다고 가정하자. 그림에서 원은 각 무선 단말을, 선은 단말 간의 링크를, 선 옆의 숫자는 해당 링크의 전송 속도 (Mbps)를 나타낸다. 우선, 단말 A 는 (b)와 같이 모든 1 홉 이웃들의 목록을 in-set 목록에 저장한다. 그리고, 가장 전송 속도가 낮은 링크, 즉, 단말 E 와 F 를 찾아 해당 단말을 목록에서 제거한 후, 내부 단말로부터 더 빠른 시간에 메시지 전송이 가능한지 확인한다. 이때, 단말 E 는 단말 B 를 통해 도달 가능하며, 상대적인 전송 시간이 $1/5.5 + 1/5.5$ 이므로 단말 A 로부터 직접 단말 E 로 직접 전송하는 경우의 전송 시간인 $1/1$ 에 비해 짧은 시간에 전송이 가능하다. 단말 F 의 경우에는 $1/11 + 1/11$ 로 더 짧은 시간에 전송이 가능하다. 따라서, (c)와 같이 단말 E, F 를 out-set 로 이동시킬 수 있다. 이 과정을 반복하여, (d)와 같이 단말 B 를 in-set 에서 제외하는 경우, in-set 내에서 단말 B 로 5.5 Mbps 보다 더 짧은 시간에 전송 가능한 경로가 존재하지 않게 되어, 이 시점에 알고리즘은 종료된다.

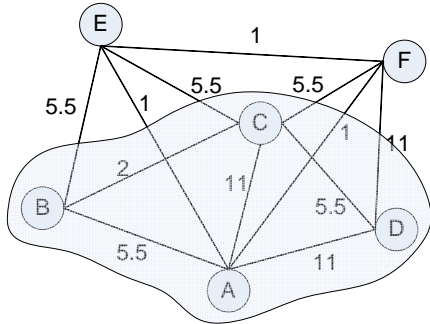
최종적으로 5.5 Mbps 가 전송 속도로 결정되어, 단말 A 는 5.5 Mbps 로 브로드캐스트를 수행한다. 이를 받은 단말 B, C, D 에서 동일한 알고리즘을 수행하여 플러딩이 동작하게 되며, 이때, 발신 MAC 주소와 시퀀스 넘버 비교, 이웃 목록 비교를 통해 릴레이를 수행할 지를 판단하는 과정이 선행된다.



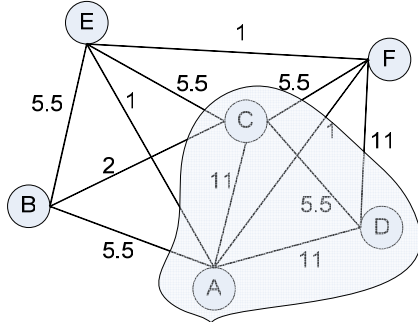
(a) Example network topology



(b) Initial in-set coverage



(c) In-set coverage after removing 2 nodes



(d) Final in-set coverage

그림 2. 제안한 알고리즘의 전송 속도 결정 예

4. 성능 평가

4.1 수치적 분석

보다 명료한 분석을 위해 그림 3 과 같은 격자 (grid) 형태의 네트워크 구성을 가정한다. 격자 간격은 50m 이며, 아래 그림과 같이 상하로 인접한 단말의 경우 11 Mbps 로 전송 가능하며, 대각선에 위치한 단말이나 다음 격자 선에 위치한 단말의 경우 2 Mbps 로 도달 가능하다고 가정한다. 자체 실험 결과, Orinoco 사의 IEEE 802.11b NIC 의 경우, 실내에서 11 Mbps 의 전송 거리는 50 m, 2 Mbps 의 전송 거리는 90 m 로 측정되었다.

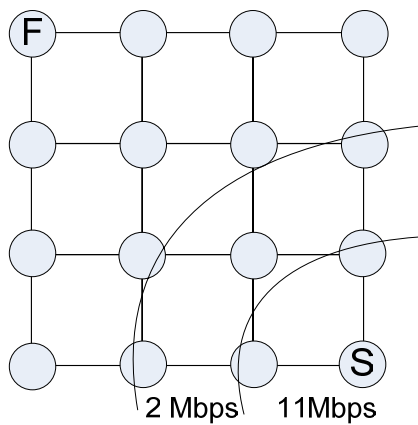


그림 3. 네트워크 토폴로지 예

현재 일반적인 IEEE 802.11b 제품의 경우, 브로드캐스트 시 basic rate set 중 2 Mbps 의 전송 속도를

선택하며, 그림의 토폴로지에서 제안 알고리즘을 수행하는 경우 11 Mbps 가 선택된다. 그림에서 단말 S 에서 2 Mbps 로 플러딩 하는 경우, 가장 멀리 위치한 단말 F 에 도달하는데 총 3 홵이 소요된다. 같은 토폴로지에서 11 Mbps 로 플러딩 하는 경우 총 6 홵이 소요된다.

따라서, 2 Mbps 전송 속도로 1 Kbits 를 플러딩 하는데 소요되는 시간은 다음과 같다.

$$\frac{1 \text{ Kbits}}{2 \text{ Mbps / second}} \times 3 \text{ hops} = 1.5 \text{ msec}$$

이에 반해, 11 Mbps 로 1 Kbits 를 플러딩 하는데 소요되는 시간은 다음과 같다.

$$\frac{1 \text{ Kbits}}{11 \text{ Mbps / second}} \times 6 \text{ hops} = 0.545 \text{ msec}$$

결과적으로, 충돌이나 헤더로 인한 오버헤드를 고려하지 않는 경우, 앞의 그림과 같은 토폴로지에서 약 1/3 정도로 전송 시간 단축을 기대할 수 있다.

4.2 모의 실험 환경

모의 실험은 NS-2 [13]를 사용하여 수행하였다. 실험에 사용한 네트워크 구성은 앞의 분석에 사용한 격자 형태의 토폴로지 (그림 3)와 동일한 형태로 구성하였다. MAC 계층에서 플러딩을 수행하였으며, 거리에 따라 전송 속도를 달리 하도록 조절하였다. MAC 프레임의 크기에 따라 플러딩 되는데 소요되는 시간을 측정하여 기본전송속도 플러딩 기법을 사용한 경우의 경우와 비교하여 제안한 알고리즘의 성능 평가를 수행하였다. 플러딩 소요 시간 측정은 최초 프레임이 전송된 시간과 네트워크 상의 모든 단말이 해당 프레임을 모두 받은 시점까지의 시간을 측정하는 방법을 사용하였다.

4.3 모의 실험 결과

모의 실험을 수행하여 얻은 플러딩 소요 시간 결과는 그림 4 와 같다. 기본 전송 속도를 사용하는 경우와 다중 전송 속도를 사용하는 경우 모두 프레임 크기 변화에 따라 플러딩에 소요되는 시간이 증가하는 양상을 보였다.

프레임 크기가 200 바이트 이하인 경우, 다중 전송 속도를 사용하는 경우에 오히려 기본 전송 속도를 사용하는 경우보다 더 오랜 시간이 소요되었다. 이는 높은 전송 속도를 사용하여 얻은 시간 상의 이득보다 헤더와 전송 횟수로 인한 오버헤드가 더 크기 때문으로 판단된다. 즉, 전송 횟수가 증가하여 백오프로 인한 오버헤드가 상대적으로 커지며 또한 헤더의 경우 기본 속도로 전송되는데 이로 전송 횟수가 증가하여 이로 인한 오버헤드 역시 상대적으로 크기 때문에 기본 전송 속도를 사용하는 플러딩 기법보다 긴 시간이 소요되었다.

그러나, 200 바이트를 초과하는 메시지의 경우, 기본 전송 속도를 사용하는 경우에 비해 빠른 시간 내에 플러딩이 수행되며, 또한 메시지의 크기가 증

가함에 따라 격차가 벌어지는 것을 확인할 수 있다. 이는 메시지의 크기가 증가함에 따라 헤더의 오버헤드가 상대적으로 줄어들기 때문이다. 프레임의 크기가 2000 바이트인 경우, 기본 전송 속도 플러딩은 31 msec의 시간이 소요되는데 반해, 다중 전송 속도 플러딩은 17 msec의 시간이 소요되었다. 이는 약 47%의 시간 이득에 해당한다.

앞의 분석 결과에서 약 66%의 시간 상의 이득을 예상하였으나 실험 결과 47% 정도의 이득을 얻은 이유로 크게 다음의 두 가지를 들 수 있겠다. 첫째, 홑 수가 증가함에 따라 간섭에 의한 영향이 더 커진 것을 들 수 있다. 높은 전송 속도를 사용하여 유효 전송 거리가 줄어들어 따라 필요한 전송 회수가 증가하였고 이에 따라 추가적인 시간이 소요되었다. 둘째, 물리 계층 헤더가 MPDU 전송 속도와 무관하게 낮은 속도로 전송되어 예상보다 시간이 더 소요되었다. 그러나, 47% 정도의 시간 향상은 상당히 의미 있는 결과라 하겠다.

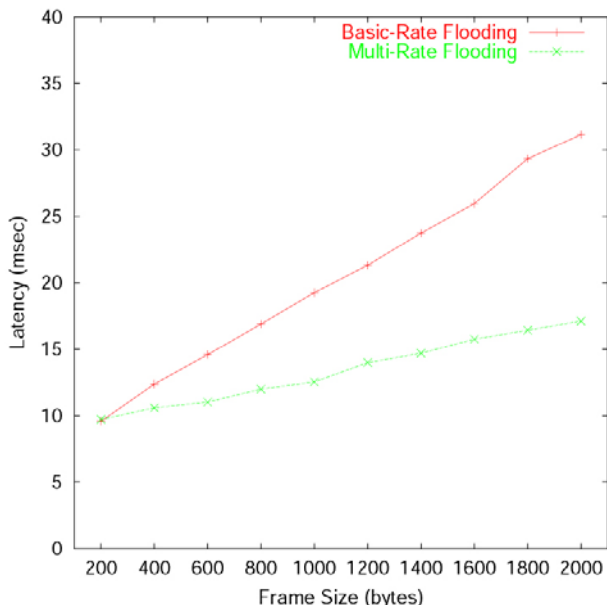


그림 4. 프레임 크기에 따른 플러딩 소요 시간 비교

5. 결론

본 논문은 MANET 환경에서 다중전송속도 특성을 고려한 플러딩 알고리즘을 소개하였다. 브로드캐스트 시 기본 전송 속도를 사용하는 기존 방법의 경우, 브로드캐스트 전송 자체의 소요 시간도 길어질 뿐만 아니라 전체 시스템 효율의 감소를 유발한다. 본 연구에서 제안한 알고리즘은 각 단말의 링크 상황에 따라 최적의 전송 속도를 선택하여 브로드캐스트를 수행하여 전송 속도 면에서 큰 이득을 보인다. NS-2를 사용한 모의 실험 분석 결과는 이를 증명한다.

6. 참고 문헌

- [1] IEEE Computer Society. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 1997.
- [2] Y. Seok, J. Park and Y. Choi, "Multi-rate Aware Routing Protocol for Mobile Ad Hoc Networks," in Proc. of IEEE VTC 2003.
- [3] Hao Zhu and Guohong Cao, "rDCF: A Relay-enabled Medium Access Control Protocol for Wireless Ad Hoc Networks," in Proc. of IEEE Infocom 2005.
- [4] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath. Flooding for Reliable Multicast in Multi-hop Ad Hoc Networks. In Proc. of the Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication, pages 64-71, 1999.
- [5] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson. A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks. Internet Draft: draft-ietf-manet-simple-mbcast-01.txt, July 2001.
- [6] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In Proc. Of MOBICOM'99, pages 151-162, 1999.
- [7] Y. Tseng, S. Ni, and E. Y. Shih. Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Networks. In Proc. of ICDCS'01, pages 481-488, 2001.
- [8] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In Swiss Federal Institute of Technology, Technical Report IC/2002/54, 2002.
- [9] Ying Cai, Kien A. Hua, and Aaron Phillips. Leveraging 1-hop Neighborhood Knowledge for Efficient Flooding in Wireless Ad Hoc Networks. 24rd IEEE Int'l Performance, Computing, and Communication Conference (IPCCC'05).
- [10] H. Lim and C. Kim. Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks. In Proc. of the ACM Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), pages 61-68, August 2000.
- [11] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In Proc. of MOBICOM'01, pages 85-96, July 2001.
- [12] W. Peng and X. Lu. On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks. In Proc. of MOBIHOC'00, 2000.
- [13] "Network Simulator," <http://www.isi.edu/nsnam/ns>.