

해상 MANET 을 위한 항로 기반 버텍스 중심 라우팅 프로토콜*

이윤도**, 이호진**, 권태경**, 최양희**, 손주영***
서울대학교 컴퓨터공학부**, 한국해양대학교 IT 공학부***

{yldlee, lumiere, tk, yhchoi}@mmlab.snu.ac.kr**, mmlab@mail.hhu.ac.kr***

VBV: Course-based Vertex-By-Vertex Routing Protocol for MANET at Sea*

Youndo Lee**, Hojin Lee**, Taekyoung Kwon**, Yanghee Choi**, Jooyoung Son***
School of Computer Science and Engineering, Seoul National University**
Division of Information Technology, Korea Maritime University***

요 약

바다 위에서 인터넷을 사용하기 위해서는 많은 비용이 들고 품질도 육상에 비해 많이 떨어진다. 현재의 망으로는 저비용 고품질의 서비스를 제공하는 것이 불가능하다. 본 논문에서는 해상에 적합한 망 모델로 MANET 을 제시한다. MANET 을 구성하기 위해서는 라우팅 프로토콜이 필수적으로 개발되어야 한다. MANET 라우팅 프로토콜은 이미 많이 연구되었지만 대부분의 연구는 노드의 분포가 균일한 상황을 가정하고 진행되었다. 그러나 바다에서는 이러한 가정이 성립하지 않기 때문에 새로운 연구가 필요하다. 본 논문에서는 항로를 이용하는 새로운 MANET 라우팅 프로토콜을 제안하고 기존 라우팅 프로토콜과의 비교를 통하여 성능향상을 보인다.

1. 서론

현재 해상에 존재하는 선박들은 현실적으로 인터넷을 사용할 수 없는 환경을 가지고 있다. 해상에도 MF, HF, VHF 라디오 통신 혹은 해사통신위성(Inmarsat)을 이용하는 통신망이 구축되어 있지만 제공할 수 있는 대역폭이 매우 좁고(64~300Kbps) 이용료도 상당히 비싸기 때문에 인터넷 접속을 목적으로 기존의 통신망을 사용하는 것은 불가능하다고 해도 과언이 아니다.

현실적인 대안으로 선박에 이동전화 기지국을 설치하고 WCDMA 방식을 이용하여 인터넷 서비스를 제공하는 방식이 고안되었으나 이 방식은 지원 가능한 대역폭이 64Kbps~2Mbps 에 불과하기 때문에 높은 품질의 서비스를 제공할 수가 없다는 문제점을 가지고 있다. 또 다른 대안으로는 항구에 무선 LAN 의 AP 를 설치하고, 항구 근처에 있는 선박들에게 무선 LAN 을 통하여 인터넷 서비스를 제공하는 방식이 있다. 이 방식은 넓은 대역폭을 지원하기 때문에 비교적 높은 품질의 서비스를 제공할 수 있지만 서비스 범위가 AP 의 전송 범위로 한정되기 때문에 실효성이 떨어진다.

위에서 제기되었던 문제점들을 효과적으로 해결하기 위해서는 새로운 방식의 해상통신망이 구축되어야 한다. 해상에는 고정체가 없다는 점, 선박들이 계속해서 움직인다는 점, 인프라스트럭처를 구축하기 어렵다는 점 등을 고려해볼 때 MANET [1]이 해상에 가장 적합한 망 모델이라고 할 수 있다. 해상에서 MANET 을 구성하기 위해서는 주소 부여, 인터넷 게이트웨이와의 연결, 라우팅 등의 문제가 존재하지만 본 연구에서는 우선적으로 라우팅 프로토콜에만 초점을 맞추고 있다.

MANET 라우팅 프로토콜은 그 특성을 기준으로 크게 3 가지로 나눌 수 있다 [1].

- Proactive 라우팅 - 사전에 모든 노드에 대한 경로를 파악한다
- Reactive 라우팅 - 필요한 시기에 필요한 경로만 파악한다
- Geographic 라우팅 - 노드의 위치정보를 이용하여 목적지에 가장 가까운 이웃에게 패킷을 전달한다. 따라서 경로 탐색 과정이 존재하지 않는다

Geographic 라우팅 방식은 노드의 위치 정보를 알아야 한다는 단점이 있지만 다른 방식들에 비해 오버헤드 및 유지해야 할 정보를 줄일 수 있다는 장점이 있다. 해상에서는 각각의 선박들이 노드가 되는데 선박들은 기본적으로 GPS 를 탑재하기 때문에 추가 비용 없이 자신의 위치를 쉽게 파악할 수 있으므로 해양에서의 MANET 은 Geographic 라우팅 방식을 이용하는 것이 유리하다. 이미 다수의 Geographic 라우팅 프로토콜들(GEDIR, MFR, GPSR) [2, 4]이 제안되어 있지만 이 프로토콜들은 노드가 균일하게 분포되어 있다는 가정을 한다. 만약 노드의 분포가 균일하지 않다면 홀(홀이란 현재 노드보다 목적지에 더 가까운 이웃 노드가 존재하지 않아서 더 이상 라우팅이 진행되지 못하는 상황이다.)이 빈번히 발생해서 오버헤드가 매우 커지게 된다. 그런데 선박들은 항로를 따라서 이동하기 때문에 그 분포가 균일하지 않다. 따라서 기존의 프로토콜을 그대로 해상에서의 MANET 에 적용하는 것은 문제가 있다. 그래서 본 논문에서는 해상에서의 MANET 에서 효율적으로 동작할 수 있는 새로운 Geographic 라우팅 프로토콜(VBV, Vertex-By-Vertex 라우팅)을 제안한다.

논문의 구성은 다음과 같다. 다음 장에서 제안된 프로토콜의 큰 틀을 설명하고 3 장에서 제안된 라우팅 프로토콜을 자세히 설명한 후에 4 장에서 성능 분석 결과를 제시하고 5 장에서 결론을 내린다.

2. 프로토콜 개관

선박들은 항로를 따라서 움직이기 때문에 항로를 잘 이용하면 효율성을 높일 수 있다. 제안된 프로토콜에서는 항로의 집합을 하나의 그래프로 본다. 각 항로 간 교차점이나 항로의 끝이 vertex(정점)가 되고 항로에 의해 서로 나누는 구간들은 edge(선분)가 된다. 그림 1 은 항로의 집합을 그래프로 보는 방법을 나타낸다. 이렇게 생성된 그래프에 Floyd 알고리즘 [5]을 적용하여 모든 vertex 들 사이의 최단 경로를 계산한다. 라우팅은 이 최단경로를 이용하여 이루어진다. 모든 선박들은 항해도를 통해서 전체 항로를 알 수 있고 항로는 정적인 정보이기 때문에 각각의 선박이 Floyd 알고리즘을 사전에 한 번만 적용하면 모든

* 본 논문은 2006 년도 한국과학재단(KOSEF)의 국제협력 연구 사업, 두뇌한국 21 지원을 받아 수행되었음

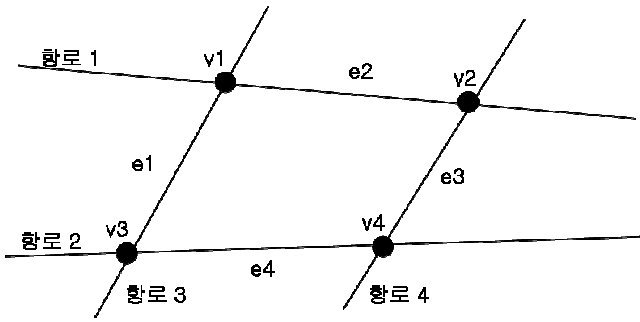


그림 1. 검은 점(v1, v2, v3, v4)가 vertex 가 되고 검은 점을 잇는 선(e1, e2, e3, e4)가 edge 가 된다.

선박들은 임의의 vertex 들 사이의 최단 경로를 알 수 있다. 각각의 선박에서 Floyd 알고리즘이 한 번만 수행되므로 이 과정의 오버헤드는 무시할 수 있다. 모든 선박들이 임의의 vertex 들 사이의 최단 경로를 알고 있으므로 라우팅이 최단 경로를 통하여 이루어질 수 있다. 문제는 일반적으로 항로그래프 상의 edge 는 실제로 선박들 간의 통신에서는 1 홉이 아니라는 점이다. 항로그래프 상의 한 vertex 에서 이웃 vertex 로 패킷이 전달될 때 실제로는 여러 선박을 거쳐야 한다. 한 edge 상에서의 패킷 전달은 Greedy Forwarding 방식 [4]을 사용한다. 각각의 선박들은 자신의 이웃 선박 중에서 다음 vertex 에 가장 근접해 있는 선박에게 패킷을 전달하고 여러 선박을 거쳐서 다음 vertex 에 도달하면 항로그래프 상의 최단경로를 이용하여 새로운 다음 vertex 가 정해지고 이 vertex 를 향하여 Greedy Forwarding 이 다시 일어난다. 이 과정을 반복적으로 수행하면 결국에는 목적지에 도착하게 될 것이다. 프로토콜의 자세한 알고리즘은 다음 장에서 설명한다.

3. VBV 라우팅 프로토콜

프로토콜은 크게 4 가지 부분으로 나눌 수 있다. 첫 째는 라우팅 패킷 헤더, 둘째는 패킷 전달, 셋 째는 홀 처리, 마지막은 헬로 패킷의 교환이다.

3.1 라우팅 패킷 헤더

라우팅 패킷 헤더는 여러 필드로 구성되어 있다. next vertex id(4byte)는 항로그래프 상에서의 다음 vertex 를 의미한다. destination vertex id(4byte)는 목적지 노드(즉, 배)가 위치해 있는 edge 의 양 끝 vertex 중 하나를 의미한다. previous vertex 는 직전에 지나쳐 온 vertex 를 의미한다. source location(8byte)은 소스 노드의 지리적 위치이고 destination location(8byte)은 목적지 노드의 지리적 위치이다. m(1bit)은 mode bit 으로 Greedy 모드와 Perimeter 모드를 구분한다. 각각의 모드에 대한 설명은 3.2 와 3.3 에 나와 있다. i(1bit)는 inconsistent bit 으로 이에 대한 설명은 3.3 에 나온다. hole #(6bit)은 홀의 개수를 의미하고 hole list(4byte*Hole #)는 홀 vertex 들의 id(4byte*N)를 의미한다. 홀에 대한 설명은 3.3 에 나와 있다. 여기서는 패킷이 처음에 생성될 때 next vertex id 와 destination vertex id 를 선정하는 방법을 설명한다. 패킷을 보내려는 노드가 위치한 edge 의 양 끝 vertex 를 s_1, s_2 라 하고 패킷의 목적지 노드가 위치한 edge 의 양 끝 vertex 를 d_1, d_2 라 하면 $[(\text{소스와 } s_1 \text{ 사이의 거리}) + (s_1 \text{ 와 } d_1 \text{ 사이의 거리}) + (d_1 \text{ 와 목적지 사이의 거리}), i, j \in \{1, 2\}]$ 가 최소가 되는 s_i 가 next vertex 가 되고 d_j 가 destination vertex 가 된다. 4 가지 경우 중에서 그 거리가 가장 짧은 경로를 선택하는 것이다. next vertex id 는 패킷이 항로그래프 상의 vertex 에 도달할 때마다 변경되

며 destination vertex 에 도착하게 되면 next vertex id 는 미리 설정된 값(실제 vertex 에 사용되지 않는 id, 본 논문에서는 0 을 사용한다.)으로 할당된다. source location 에는 GPS 를 이용하여 측정한 자신의 지리적 위치를 써넣고 destination location 에는 목적지 노드의 위치를 써넣는다. 목적지 노드의 위치는 노드의 주소를 위치로 매핑하는 위치 등록 및 검색 서비스를 통하여 알 수 있다고 가정한다.[6]

3.2 패킷 전달

패킷은 항로그래프 상의 최단 경로를 통하여 전달되는데 항로그래프 상의 edge 는 실제로는 멀티 홉일 수 있기 때문에 edge 상에서도 라우팅이 필요하다. edge 상에서의 라우팅은 다음 vertex 를 목적지로 하는 Greedy Forwarding 방식을 사용한다. 패킷 전달 알고리즘 기술에 이용되는 함수들의 정의는 아래와 같다.

near(vertex id) : vertex 가 자신의 전송범위 내에 위치하면 true 를, 아니면 false 를 리턴한다.(vertex id 가 0 인 경우 항상 false 를 리턴)

local_maximum(vertex id) : 자신이 next vertex(혹은 목적지 노드)에 가장 가까이 위치하면 true 를, 아니면 false 를 리턴한다.

greedy_forwarding(vertex id) : vertex id 를 향하여 greedy forwarding 을 한다.(vertex id 가 0 인 경우 destination location 으로 greedy forwarding 을 한다.)

reachable(vertex id) : vertex id 로 도달가능하면 true 를, 그렇지 않으면 false 를 리턴한다.

노드가 패킷을 받았을 때 near(next vertex id)가 false 이면 next vertex 를 향하여 greedy forwarding 을 수행하고 true 이면 알고리즘 1 을 수행한다. 알고리즘 1 에서는 먼저 next vertex id 와 destination vertex id 를 비교하여 그 값이 같을 경우에 현재의 노드가 next vertex 에 가장 가깝다면 next vertex id 를 0 으로 할당하고 destination location 을 향하여 greedy forwarding 을 수행하고 그렇지 않다면 next vertex id 를 바꾸지 않고 greedy forwarding 을 수행한다. next vertex id 와 destination vertex id 가 같지 않은 경우의 동작은 mode bit 의 값에 따라 두 경우로 나뉜다. perimeter 모드에서는 GPSR face routing 을 하고 greedy 모드에서는 greedy forwarding 을 한다. perimeter 모드에 대한 추가적인 설명은 3.3 에 나온다. greedy 모드의 경우에는 항로그래프 상의 최단경로를 보고 next vertex id 를 최단경로 상의 다음 vertex 로 변경한 이후에 greedy forwarding 을 한다. 이 때 next vertex id 를 변경하기 전에 next vertex 가 도달가능한지를 확인하는데 이런 과정이 필요한 이유는 항로그래프 상의 edge 는 여러 선박들로 연결되는 것이어서 edge 상에 선박들이 띄엄띄엄 위치해 있으면 next vertex 까지 패킷이 전달되지 못할 수도 있기 때문이다. next vertex 가 도달가능하지 않다면 홀이 발생한 것이므로 홀 처리를 해주어야 한다. 도달가능여부를 확인하기 위해서는 다른 vertex 들의 정보까지 알아야 하기 때문에 헬로 패킷을 서로 교환하여 Reachability table 을 관리하게 된다. 이에 대한 자세한 설명은 3.4 에 나와 있다.

3.3 홀 처리

```

If (next vertex id == dst vertex id)
  if(local_maximum(next vertex id))
    next vertex id <- 0
    greedy_forwarding(next vertex id)
  else
    greedy_forwarding(next vertex id)
else
  if (mode == perimeter)
    if (local maximum에서 탈출)
      mode == greedy
    else
      GPSR(face routing) 처리
  if(mode == greedy)
    tmp vertex id <- 최단 경로 상의 next vertex id
  if (reachable(tmp vertex id))
    next vertex id <- tmp vertex id
    greedy_forwarding(next vertex id)
  else
    if(local_maximum())
      hole 발생 => hole 처리
    else
      greedy_forwarding(next vertex id)

```

알고리즘 1. near(next vertex id)가 true 일 경우 실행되는 알고리즘

홀은 두 가지 경우에 발생할 수 있다. 첫 번째는 3.2 에 나왔던 내용처럼 패킷 헤더 상의 next vertex 에 도달하여 최단경로를 보고 다음 vertex 를 정할 때 다음 vertex 가 도달가능하지 않아서 홀이 발생하는 경우이고 두 번째는 next vertex 로의 greedy forwarding 을 하던 도중에 next vertex 에 도달하지도 못하고 local maximum 에 빠지는 경우이다. 홀이 발생하게 되면 패킷 헤더의 Hole# 필드 값을 1 증가시키고 홀이 발생한 vertex 의 id 를 홀 리스트에 추가한다. 첫 번째 경우의 홀은 최단 경로의 edge 상에 배들이 많지 않아서 edge 의 연결성이 보장되지 못해 생기는 것이다. 따라서 최단 경로 상의 vertex 가 아닌 다른 vertex 를 선택하면 홀에서 빠져나올 수 있다. 단 다른 vertex 를 통한 경로의 연결성이 최대한 보장되어야 하므로 세 가지 조건을 확인해야 한다. 먼저 현재 vertex 에서 선택된 vertex 까지의 연결성이 보장되어야 한다 그리고 선택된 vertex 는 홀 리스트에 포함되어 있지 않아야 한다. 마지막으로 선택된 vertex 를 통하여 destination vertex 로 가는 경로 중에 홀 vertex(홀 리스트에 포함되어 있는 vertex 들)가 포함되어있어서는 안 된다. 이 세 가지 조건을 만족하는 vertex 들 중에서 경로의 길이가 가장 짧은 vertex 를 next vertex 로 선택한다. 조건을 만족하는 vertex 가 존재하지 않는 경우에는 mode bit 을 perimeter mode 로 바꾸고 GPSR face routing [4]을 수행하게 된다. face routing 은 항로그래프 상의 edge 단위로 수행한다. 반면에 GPSR 은 face routing 을 노드 단위로 수행한다. face routing 을 노드 단위로 수행하면, local maximum 에서 일시적으로 벗어나더라도, 노드 자체가 코스상에 위치하기 때문에 다시 local maximum 에 빠질 가능성이 높다. 따라서 VBV 의 face routing 이 보다 더 효율적이다.

두 번째 경우의 홀은 두 가지로 나뉘어지는데 첫 번째는 next vertex id 가 0 인 경우이고 두 번째는 next vertex id 가 0 이 아닌 경우이다. next vertex id 가 0 인 경우는 destination vertex 에 도착해서 목적지 노드로 greedy forwarding 을 하는 도중에 홀이 발생한 경우로 이 경우에는 destination vertex 를 변경한 후에 greedy forwarding 을 해야 한다. next vertex id 가 0 이 아닌 경우에는 도달가능여부에 대한 판단이 잘못된 경우로 헬로

패킷의 교환에 의해 만들어지는 Reachability table 의 정보가 불일치해서 홀이 발생하는 것이다. 이 때에는 패킷 헤더에 적혀있는 inconsistent bit 의 값을 1 로 설정한다. 그리고 next vertex id 를 현재 previous vertex 로 설정하고 변경된 next vertex id 를 hole list 에 추가한 후 greedy forwarding 을 수행한다. inconsistent bit 가 1 인 패킷을 받은 노드들은 자신의 Reachability table 에 잘못된 정보가 있는 것이므로 Reachability table 과 Neighbor table 을 갱신해야 한다. Neighbor table 관련 내용은 3.4 에서 설명된다.

3.4 헬로 패킷

헬로 패킷의 헤더는 id(4byte), location(8byte), reachable vertex id1(4byte), reachable vertex id2(4byte)로 구성된다. 헬로 패킷을 보내는 노드는 id 에 자신의 주소를, location 에 자신의 위치를 써넣는다. Reachable vertex id1, 2 에는 자신이 위치한 edge 의 양 끝 vertex 중에서 도달가능한 vertex 를 적는다. vertex 가 자신의 전송범위 내에 있거나 Reachability table 안에 있으면 그 vertex 는 도달가능한 것이다. 헬로 패킷은 [0.5B, 1.5B](B:Hello Interval)사이의 간격을 두고 주기적으로 전달된다 [4]. 모든 노드는 2 개의 테이블을 관리한다. 하나는 neighbor table 이고 다른 하나는 Reachability table 이다. neighbor table 은 자신의 이웃 노드를 관리하는 테이블로써 각각의 엔트리는 헬로 패킷의 내용과 일치하며 각 필드의 의미는 “id-이웃노드의 Id, location-이웃 노드의 위치, reachable vertex-이웃노드에서 도달가능한 vertex 들” 이 된다. 모든 엔트리에는 타이머가 설정되어 있어서 일정 시간 동안 헬로 패킷을 받지 못하면 자동으로 테이블에서 삭제된다. Reachability table 은 vertex 에 대한 도달가능여부를 관리하는 테이블로 각각의 엔트리는 vertex id(4byte), next hop1(4byte), next hop2(4byte), ... 로 구성된다. 각각의 엔트리는 vertex id 로 next hop1, next hop2, ...을 통하여 도달가능하다는 것을 의미한다.(next hop 은 자신의 이웃 노드 중 해당 vertex id 에 도달가능한 노드들로 채워진다.) Reachability table 은 neighbor table 로부터 생성된다. neighbor table 에 새로운 reachable vertex 가 추가되면 reachability table 에도 새로운 엔트리 혹은 next hop(이미 해당 vertex id 가 테이블에 존재하는 경우)이 추가되고 neighbor table 에서 reachable vertex 가 삭제되면 reachability table 의 next hop 이 삭제된다. next hop 이 모두 삭제되면 해당 vertex 는 도달가능하지 않은 것이므로 엔트리 전체가 삭제된다. Reachability table 에서 엔트리가 추가되거나 삭제되면 도달가능여부 정보가 바뀔 것이기 때문에 이 때에는 헬로 패킷을 주기와 상관없이 보내서 다른 노드의 Reachability table 이 최신정보를 유지할 수 있게 해야 한다.

4. 성능 평가

이 장에서는 VBV 의 성능향상을 알아보기 위해서 기존의 라우팅 프로토콜(AODV [3] - reactive, GPSR [4] - geographic)과 비교 실험을 했다. 시뮬레이터는 NS-2 [7]를 사용했다. MAC 프로토콜은 NS-2 에 구현되어 있는 802.11 [8]이며 대역폭은 1 Mbps 이고 전송 범위는 250m 이다. 실제로 배에서 사용될 MAC 프로토콜은 802.11b 가 아닐 가능성이 높다. 왜냐하면 더 높은 대역폭과 더 넓은 전송 범위를 필요로 하기 때문이다. 그 적합한 예로 802.16 이 있는데, 현재 802.16 이 NS-2 에 구현되어 있지 않아서 우리는 802.11b 를 사용했으며 그 에 따라서 노드 배치와 노드 분포 영역을 비례해서 설정했다. 2000m x 2000m 의 정사각형의 각 변에 임의로 6 개의 항구를 배치시키고 각 항구를

임의로 선택해서 8 개의 코스를 만들었다. 노드는 200 개를 코스 위에 분포시켰으며 각 노드는 코스를 따라서 max speed 이하의 임의의 속도로 이동한다. 코스의 끝(항구)에 도달하면 그 끝에 연결된 코스 중 하나를 임의로 고른 후 다시 이동을 시작한다. 트래픽은 CBR 을 사용했으며 패킷 크기는 64 바이트, 전송 간격은 1 초이다. 총 10 개의 세션을 만들었으며 소스와 목적지는 임의로 선택된다.

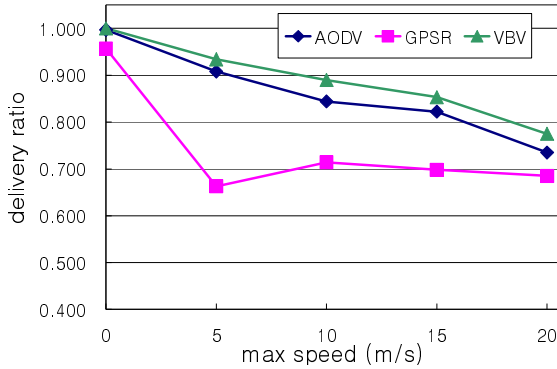


그림 2. 이동성에 따른 delivery ratio

그림 2 는 이동성에 따른 delivery ratio 를 나타낸다. 이동성이 높아질 수록 모든 라우팅 프로토콜의 delivery ratio 가 감소하게 된다. 이는 각 프로토콜들이 유지하는 라우팅 정보가 이동성에 의해서 실제 정보와 일치하지 않기 때문이다. 즉, 노드의 이동으로 라우팅 프로토콜의 정보상으로는 연결된 링크이지만 실제적으로 끊기게 되기 때문이다. 모든 경우에 있어서 VBV 의 delivery ratio 가 가장 우수함을 알 수 있다. AODV 는 라우팅 테이블에 경로가 없으면 route discovery 과정을 통해서 경로를 찾지만 노드의 이동으로 그 경로가 끊기게 되면 패킷들이 드랍되게 된다. 이에 비해서 geographic routing 은 별도의 경로 탐색 과정이 존재하지 않고 패킷을 받을 때마다 이웃노드의 정보를 이용해서 포워딩을 한다. 그렇기 때문에 VBV 는 이동성이 높아 지더라도 AODV 에 비해서 더 높은 delivery ratio 를 보인다. GPSR 은 노드가 코스 상에 분포하기 때문에 hole 에 빈번히 빠진다. 그렇기 때문에 face routing 을 빈번히 수행하게 되고 그 결과 목적지의 주변을 돌아가는 경우가 많아지고 그 도중에 collision 이나 TTL limit 으로 드랍이 자주 발생한다.

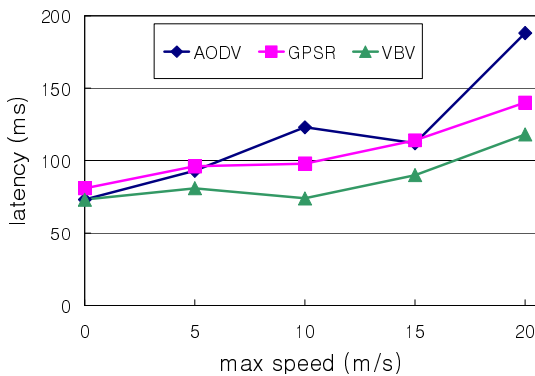


그림 3. 이동성에 따른 latency

그림 3 은 이동성에 따른 latency 를 나타낸다. AODV 는 경로를 찾을 때 별도의 경로 획득 과정을 필요로 하므로 추가 delay 가 필요하다. 반면에 geographic 라우팅은 별도의 경로 획득 과정이 없기 때문에 추가 delay 가 존재하지 않는다. 따라서 VBV 의

latency 가 가장 낮다. 하지만 GPSR 은 hole 에 자주 빠지기 때문에, face routing 을 자주 수행하게 되고 그 결과 목적지를 돌아서 찾아가게 되므로 latency 가 상당히 커지게 된다.

5. 결론

우리는 이 논문에서, 해상에서 배들이 코스상에 배치되고 코스 정보는 정적인 정보로 각 배들이 모두 알고 있다는 사실과 각 배들은 GPS 를 탑재하는 것을 이용해서 지리정보를 이용한 VBV 라우팅 알고리즘을 제안했다.

지리정보를 이용할 수 있는 경우 geographic routing 을 사용하면 오버헤드를 줄이고, latency 를 줄일 수 있는 점 등 많은 이점이 있지만, 이는 노드가 임의로 균일하게 분포하는 경우에 해당된다. 해상에서의 배들처럼 노드가 어떤 코스 상에 존재하는 경우, 일반적인 geographic routing 을 사용하면 hole 에 빈번히 빠지게 되기 때문에 오히려 delivery ratio 와 latency 가 일반적인 reactive routing 에 비해서 저하된다.

시뮬레이션을 통해서 VBV 가 AODV, GPSR 에 비해서 delivery ratio, latency 모두 더 나은 성능을 보임을 확인했다.

참고 문헌

- [1] I. Chlamtac, M. Coti, J J-N. Liu, " Mobile adhoc networking : imperatives and challenges ", *Elsevier Ad Hoc Networks*, vol. 1, pp. 13~64, 2003
- [2] I. Stojmenovic and X. Lin, " Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks, " *IEEE Trans. Parallel Distributed Systems*, vol. 12, no. 10, 2001.
- [3] C. Perkins, E. Belding-Royer and S. Das, " Ad hoc On-demand Distance Vector (AODV) Routing, " *IETF Experimental RFC*, MANET working group, RFC 3561, July 2003
- [4] B. Karp and H.T. Kung, " GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, " *in Proc. ACM/IEEE MobiCom ' 00*, 2000
- [5] Robert W. Floyd. " Acm algorithm 97: Shortest path, " *Communications of the ACM*, vol. 5, no. 6, June 1962
- [6] J. Li, J. Jannotti, D. DeCouto, D. Karger and R. Morris, " A Scalable Location Service for Geographic Ad-hoc Routing, " *in Proc. ACM/IEEE MobiCom ' 00*, 2000
- [7] The Network Simulator - NS2, <http://www.isi.edu/nsnam/ns/>, Online Link
- [8] " IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, " IEEE 802.11 Standard, August 1999