



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년02월06일
(11) 등록번호 10-0882738
(24) 등록일자 2009년02월02일

(51) Int. Cl.

H04L 12/28 (2006.01) H04B 7/26 (2006.01)

(21) 출원번호 10-2007-0083411

(22) 출원일자 2007년08월20일

심사청구일자 2007년08월20일

(56) 선행기술조사문헌

논문:LNCS*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

경희대학교 산학협력단

경기도 용인시 기흥구 서천동 1 경희대학교 국제 캠퍼스내

(72) 발명자

이승룡

경기 성남시 분당구 구미동 250 삼환빌라 102동 102호

이영구

경기 수원시 영통구 영통동 1024-14 경희유니빌 706호

후이 쉬

대한민국 경기도 용인시 기흥구 서천동 1번지 경희대학교전자정보과

(74) 대리인

김삼수

전체 청구항 수 : 총 7 항

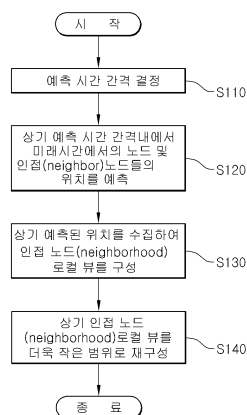
심사관 : 김현수

(54) 모바일 애드혹 네트워크의 인접노드 추적 방법

(57) 요약

네트워크 프로토콜의 설계에서 포워딩 결정을 용이하게 할 수 있도록, 모바일 노드의 인접노드(neighborhood) 로컬 뷰(local view)를 해결하는 방법이 제공된다. 일반적인 모바일 애드혹 네트워크의 노드는 Hello 메시지를 주기적으로 수신함으로써 로컬 토폴로지 뷰를 설정한다. 이 방법을 기초하여 수신된 "Hello" 메시지에서 추출된 저장 히스토리컬(historical) 정보에 기초하여 노드들의 위치를 예측하고, 이 예측된 위치들을 수집하여 인접노드(neighborhood) 뷰(view)를 구성함으로써, 토폴로지 로컬 뷰(topology local view)가 유용하게 이루어지도록 하여, 네트워크 통신이 최적의 포워드 결정들을 설정하고 통신 성능(performance)을 개선할 수 있는 일관적인 토폴로지 로컬 뷰를 제공한다는 효과가 있다.

대표도 - 도1



특허청구의 범위

청구항 1

예측 시간 간격(interval)을 결정하는 단계;

상기 결정된 예측 시간 간격 이내에 미래 시간의 상기 노드 및 인접(neighbor) 노드들의 위치를 예측하는 단계;

상기 예측된 노드들의 위치를 수집하여 인접노드(neighborhood) 로컬 뷰를 구성하는 단계;

상기 인접노드(neighborhood) 로컬 뷰를 더 작은 범위로 재구성하는 단계

를 포함하는 모바일 애드혹 네트워크에서의 인접노드(neighborhood) 추적(tracking) 방법.

청구항 2

제 1항에 있어서, 상기 예측 시간 간격은 전송 범위 체류 시간(T_{dwell})을 분석하여 얻을 수 있고,

$$E[T_{dwell}] = \frac{\pi A}{E[V]L}$$

여기서, $E[T_{dwell}]$ 은 T_{dwell} 의 평균값이고, A 는 전송 범위 영역이며, L 은 이 영역의 페리미터(perimeter)이고, $E[V]$ 는 V 의 평균값이고, V 는 노드의 상대속도벡터인 모바일 애드혹 네트워크에서의 인접노드(neighborhood) 추적(tracking) 방법.

청구항 3

제 2항에 있어서, 상기 V 의 평균값 $E[V]$ 는

$$E[V] = \frac{1}{\pi^2} \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} (v_1 + v_2) F_e \left(\frac{2\sqrt{v_1 v_2}}{v_1 + v_2} \right) f_V(v_1) f_V(v_2) dv_1 dv_2$$

이고, 여기서

$$F_e(k) = \int_0^1 \sqrt{\frac{1-k^2 t^2}{1-t^2}} dt$$

은 두 종류의 엘립틱(elliptic) 적분으로 이루어지고, v_1 은 노드의 이동 속도이고, v_2 는 인접 노드의 이동 속도이며, $f_V(v_1)$ 은 v_1 의 조인트 pdf이고, $f_V(v_2)$ 는 v_2 의 조인트 pdf인 모바일 애드혹 네트워크에서의 인접노드(neighborhood) 추적(tracking) 방법.

청구항 4

제 1항에 있어서, 상기 노드 및 인접(neighbor) 노드들의 위치를 예측하는 방법은 위치 기반 방법을 포함하며, 상기 위치 기반 방법은

$$\begin{cases} x_p = x_{1h} + \frac{x_{1h} - x_{2h}}{t_{1h} - t_{2h}} (t_p - t_{1h}) \\ y_p = y_{1h} + \frac{y_{1h} - y_{2h}}{t_{1h} - t_{2h}} (t_p - t_{1h}) \\ z_p = z_{1h} + \frac{z_{1h} - z_{2h}}{t_{1h} - t_{2h}} (t_p - t_{1h}) \end{cases}$$

이고, 여기서 (x_p, y_p, z_p) 는 미래 시간 t_p 에서의 노드의 위치이고, (x_{1h}, y_{1h}, z_{1h}) 및 (x_{2h}, y_{2h}, z_{2h}) 는 각각 t_{1h} 및 t_{2h} 시간에서의 노드의 위치이며, 이때 t_{1h} 는 t_{2h} 보다 큰 것을 특징으로 하는 모바일 애드혹 네트워크에서의 인접노드(neighborhood) 추적(tracking) 방법.

청구항 5

제 1항에 있어서, 상기 노드 및 인접(neighbor) 노드들의 위치를 예측하는 방법은 속도 보정 방법을 포함하며, 상기 속도 보정 방법은

$$\begin{cases} x_p = x_{1h} + v'_x(t_p - t_{1h}) \\ y_p = y_{1h} + v'_y(t_p - t_{1h}) \\ z_p = z_{1h} + v'_z(t_p - t_{1h}) \end{cases}$$

이고, 여기서 (x_p, y_p, z_p) 는 미래 시간 t_p 에서의 노드의 위치이고, (x_{1h}, y_{1h}, z_{1h}) 는 t_{1h} 시간에서의 노드의 위치이며, (v'_x, v'_y, v'_z) 는 특정 노드에 대한 최신 업데이트의 속도인 모바일 애드혹 네트워크에서의 인접노드(neighborhood) 추적(tracking) 방법.

청구항 6

제 1항에 있어서, 상기 노드 및 인접(neighbor) 노드들의 위치를 예측하는 방법은 등가속도 방법을 포함하며, 상기 등가속도 방법은

$$\begin{cases} x_p = x_{1h} + \frac{2v'_x + (v'_x - v''_x) \frac{t_p - t_{1h}}{t_{1h} - t_{2h}}}{2} (t_p - t_{1h}) \\ y_p = y_{1h} + \frac{2v'_y + (v'_y - v''_y) \frac{t_p - t_{1h}}{t_{1h} - t_{2h}}}{2} (t_p - t_{1h}) \\ z_p = z_{1h} + \frac{2v'_z + (v'_z - v''_z) \frac{t_p - t_{1h}}{t_{1h} - t_{2h}}}{2} (t_p - t_{1h}) \end{cases}$$

이고, 여기서 (x_p, y_p, z_p) 는 미래 시간 t_p 에서의 노드의 위치이고, (x_{1h}, y_{1h}, z_{1h}) 는 t_{1h} 시간에서의 노드의 위치이며, (v'_x, v'_y, v'_z) 는 특정 노드에 대한 최신 업데이트의 속도이며, (v''_x, v''_y, v''_z) 는 특정 노드에 대한 두 번째 업데이트의 속도인 모바일 애드혹 네트워크에서의 인접노드(neighborhood) 추적(tracking) 방법.

청구항 7

제 1항에 있어서, 상기 인접노드(neighborhood) 로컬 뷰의 더 작은 범위는

$$E[SR] = (1 - p)R_1$$

에 의하여 구해지고, 여기서 $E[SR]$ 은 더 작은 인접 노드 범위의 예상값이고, SR 은 더 작은 인접 노드 반경이며, R_1 은 노드의 전송 범위이고, p 는 특정 노드에 대하여 상대 속도 v 를 갖는 인접 노드가 특정 노드의 전송 점위로 이동할 확률인 모바일 애드혹 네트워크에서의 인접노드(neighborhood) 추적(tracking) 방법.

청구항 8

삭제

청구항 9

삭제

명세서

발명의 상세한 설명

기술 분야

<1> 본 발명은 일반적으로 무선 랜(LAN)의 통신에 관련된 것으로 모바일 애드혹 네트워크(Mobile Ad-Hoc Networks: 이하, MANETs)의 통신에 관련된 것이다. 특히, 본 발명은 그와 같은 네트워크에서 정확한 인접노드(neighbor) 로컬 뷰(local view)를 구성하기 위해 이동 장치들의 이동성을 예측하는 방법에 관련된 것이다.

배경 기술

<2> 모바일 애드혹 네트워크(MANETs)에 대해 배치된 프로토콜에서는 네트워크에 존재하는 각 노드들이 그들의 존재를 알리고 정보를 업데이트 하기 위해 "Hello" 메시지를 보낸다. 정기적인 업데이트에서, 노드별 "Hello"의 간격은 메시지 충돌을 줄이기 위해 비동기적일 수 있다. 각 노드는 그것의 인접노드(neighborhood)의 로컬 뷰(local view)(예를 들어, 1-hop 위치 정보)를 구성하기 위해 가장 최근에 수신된 "Hello" 메시지에서 그것의 인접노드(neighbor) 정보를 추출한다. 그러나, 이런 종류의 인접노드(neighborhood) 로컬 뷰(local view) 구성

기법은 두 가지의 문제점이 있다.

- <3> 1) 때 지난(outdated) 로컬 뷰: 브로드캐스트 혹은 라우팅이 "Hello" 메시지 간격 내에 일어나고 노드들이 그 간격동안 이동하는 일반적인 경우를 고려하면, 로컬화된 프로토콜의 포워드 결정(forward decisions)은 때 지난(outdated) 인접노드(neighborhood) 뷰에 기초해서 이루어지게 될 것이다.
- <4> 2) 비동기적인 로컬 뷰: 각 노드에서의 비동기적인 표본 주파수, 주기적인 업데이트 시의 비동기적인 "Hello" 간격, 그리고 조건적인 업데이트 시의 서로 다른 "Hello" 간격은 각 인접노드(neighbor)에 대해서 인접노드 로컬 뷰에서의 비동기적인 정보를 야기시킨다.
- <5> 또한, 때 지난(outdated) 그리고 비동기적인 네트워크 토폴로지 뷰(topology view)에 기초한 포워드 결정은 부정확할 수 있으므로 브로드캐스트(broadcast) 작업(task)의 부족한 적용 범위(coverage) 또는 라우트(route) 장애를 일으킬 수 있는 정보전달(delivery) 장애를 야기한다. 만일 네트워크의 형태 변화가 미리 예측될 수 있다면, 정보전달(delivery) 장애를 막거나 줄이기 위한 적절한 포워드 결정이 생성될 수 있다.
- <6> 인접노드(neighborhood) 추적(tracking)은, 네트워크 프로토콜의 설계에서 포워딩 결정을 용이하게 만들 수 있는 이동 노드의 인접노드(neighborhood) 로컬 뷰(local view)를 정확한 시기에 결정하는 작업(task)이다. 그러므로, 이것은 네트워크 프로토콜의 설계에서 중요할 수 있다.
- <7> 라우트(route) 경로 선택을 돕기 위한 정확한 토폴로지 뷰(topology view)를 유지하기 위해 시도하는 두 가지의 연구(work)가 존재한다. 첫 째로, Kim et al의 연구에서는 각 노드의 안정 구역 및 경계(caution) 구역이 GPS로부터 얻어지는 노드의 위치, 속도 및 방향 정보에 기초하여 정의되었다. 특히, 안정 구역은 이동 노드가 상대적으로 그들 각각에 가깝게 위치한 인접(neighbor) 노드들을 갖는 안정된 링크를 유지할 수 있는 영역이다. 경계(caution) 구역은 노드가 상대적으로 각각으로부터 멀리 떨어진 인접(neighbor) 노드들을 갖는 비 안정된 링크를 유지할 수 있는 영역이다. 두 번째로, Wu 및 Dai는 진부한 토폴로지 로컬 뷰를 보완하기 위한 보수적인(conservative) "두 개의 전송 반경" 방법을 제안하였다. 그러나, 모든 상기 접근들은 미리 정확한 네트워크 토폴로지 뷰(topology view)를 구성하기 위해 이동 노드들의 위치를 예측하는 것보다 단지 네트워크 토폴로지 뷰(topology view)의 오류를 보완하기 위해 시도하는 것이므로 능동적이지 못하다.

발명의 내용

해결 하고자하는 과제

- <8> 본 발명은 토폴로지 로컬 뷰의 업데이트가 유용하게 이루어지도록 하며, 일관적인 인접노드(neighborhood) 로컬 뷰를 구성하는데 목적이 있다.

과제 해결수단

- <9> 예측 시간 간격을 결정하고, 상기 예측 시간 간격 내에서 미래 시간의 노드들의 위치를 예측한 후, 이 예측된 위치를 수집하여 인접노드(neighborhood) 로컬 뷰를 구성하며, 또한 상기 인접노드(neighborhood) 로컬 뷰를 보다 작은 범위로 재구성한다.

효 과

- <10> 수신된 "Hello" 메시지에서 추출된 저장 히스토리컬(historical) 정보를 토대로 노드들의 위치를 예측하고, 이 예측된 위치들을 수집하여 인접노드(neighborhood) 뷰(view)를 구성함으로써, 토폴로지 로컬 뷰(topology local view)의 업데이트가 유용하게 이루어지도록 하며, 네트워크 통신이 최적의 포워드 결정들을 설정하고, 통신 성능(performance)을 개선할 수 있는 일관적인 토폴로지 로컬 뷰를 제공한다는 효과가 있다.

발명의 실시를 위한 구체적인 내용

- <11> 본 발명은 모바일 애드혹 네트워크에서 동시에 이동 노드의 인접노드(neighborhood)을 예측하는 방법을 제공하는 것이다.
- <12> 비동시성 문제를 처리하기 위해, 우리는 "Hello" 메시지에 정확한 송신(sending) 시간을 붙인다. "Hello" 메시지를 수신하는 노드들은 메시지 내용뿐 아니라 수신(reception) 시간을 포함할 것이다. "Hello" 메시지 내의 수신 시간 및 송신 시간을 비교함으로써, 우리는 두 노드들 간의 시간 차를 연산할 수 있다. 어떤 미래 시간 t 에서 어떤 노드 S 의 동기화된 로컬 뷰를 얻기 위해, 우리는 노드 S 를 참조 노드로써 세트하고, 그 인접노드

(neighbor)의 동기 시간 t' 을 추론한다. 업데이트된 인접노드(neighborhood) 로컬 뷰를 구성하기 위해, 우리는 그것의 미래 위치를 예측하기 위한 노드의 두 개 또는 하나의 최신 정보를 사용하는 선형 및 비선형 예측 모델들을 구분적으로 제안한다. 예측되는 인접노드(neighbor)의 위치를 모음으로써, 노드 S는 실제 전송시간에서 업데이트되고 동기화된 인접노드(neighborhood) 뷰를 구성할 수 있다.

<13> 본 발명은 동적인 접근이 라우팅 또는 브로드캐스팅 작업이 트리거될 때 예측하는 동기화된 인접노드(neighborhood) 로컬 뷰를 구성하도록 사용될 수 있다. 그리고 나서, 현재 배치된 프로토콜들의 실행을 개선시키기 위해 이 업데이트된 로컬 뷰에 기초하여 포워드 결정을 할 수 있다.

<14> 이하, 첨부한 도면을 참조하여 본 발명에 따른 인접노드(neighborhood) 추적(tracking) 방법에 대해 상세히 설명한다.

<15> 도 1은 본 발명의 실시 예에 따른 인접노드(neighborhood) 추적(tracking) 방법을 나타내는 순서도이다.

<16> 노드의 전송 범위 내에 상기 노드의 인접(neighbor) 노드들이 체류하는 시간을 분석하여 예측 시간 간격을 결정하고(S110), 상기 예측 시간 간격 내에서 미래 시간의 상기 노드 및 인접(neighbor) 노드들의 위치를 예측하고(S120), 상기 예측된 노드들의 위치를 모아서 인접노드(neighborhood) 로컬 뷰를 구성한 후(S130), 상기 인접노드(neighborhood) 로컬 뷰를 더 작은 범위로 재구성한다(S140).

<17> 비동기적이고 때 지난(outdated) 로컬 뷰의 문제점을 해결하기 위해, 본 발명은 노드 S의 실제 에미션(emission) 시간 t_b 와 브로드캐스트 지연 시간 t_d 의 합인, 같은 미래 시간 t_p (노드 S의 클럭을 갖는)에 노드 S의 위치와 그것의 모든 인접노드(neighborhood)를 예측한다. 예측되는 시간을 모음으로써, 노드 S는 업데이트되고 비동기화된 인접노드(neighborhood) 로컬 뷰를 구성할 수 있다. 지연시간 t_d 는 무선 네트워크 전송 지연 t_e 뿐 아니라 패킷 및 전송 처리 시간 t_s 를 포함한다. t_e 는 근본적으로 무선 네트워크 내에 고정되어 있는 반면 t_s 는 패킷 사이즈에 따라 변경할 수 있다.

<18> 게다가, 예측 간격은 또한 몇몇의 다른 요인들에 의해 영향을 받고, 경계(bound)를 갖는다. 그러나, 여전히 인접(neighbor) 노드의 대응하는 예측 시간을 연산하는 방법과 노드의 위치를 예측하는 방법과 같은 두 개의 문제점이 있다.

<19> 어떤 인접(neighbor) 노드 A의 예측 시간 t'_p 를 연산하기 위해, 노드 S, t'_d 를 참조하여 그것의 시간차를 연산할 수 있다. 그리고 $t'_p = t_p + t'_d$ 이다. t'_d 를 얻기 위해, 본 발명에서는 "Hello" 메시지 로컬 송신 시간 t_l 과 로컬 수신 시간 t_r 또한 포함한다. 그리고 두 노드 간의 시간차는 $t'_d = t_l - t_r + t_e$ 와 같이 연산될 수 있다. 여기서 t_e 는 무선 네트워크 전송 지연이다.

<20> 그러나, 상기 노드 S를 위한 실제 전송 시간을 작성할 때, 만일 예측 간격 이내에 인접(neighbor) 노드들이 이미 노드 S의 전송 범위 밖으로 움직인다면, 본 발명의 예측 기법은 의미 없을 것이다. 그러므로 본 발명에서는 어떤 인접(neighbor) 노드 U가 노드 S의 전송 범위에 머무는 범위 내의 시간 주기인, 전송범위 체류시간(transmission range dwell time), T_{dwell} 을 분석한다. R_{dwell} 은 그것의 전송 영역 경계 크로싱(crossing)의 비율이다.

<21> 도 2는 예측 간격에 대한 분석 모델을 나타내는 도면으로, 노드 S는 $\vec{V}_1$ 속도로 움직이고, 노드 U는 $\vec{V}_2$ 로 움직인다고 가정한 것이다. 노드 U에서 노드 S로의 상대 속도 $\vec{V}$ 는 아래의 수식과 같이 주어진다.

<22>
$$\vec{V} = \vec{V}_2 - \vec{V}_1$$

<23> $\vec{V}$ 의 절대값은 아래의 수식과 같이 주어진다.

<24>
$$V = \sqrt{V_1^2 + V_2^2 - 2V_1V_2 \cos(\Phi_1 - \Phi_2)}$$

<25> 여기서 V_1 과 V_2 는 각각 $\vec{V}_1$ 과 $\vec{V}_2$ 의 절대값이다. V 의 평균값은 아래의 수식과 같이 주어진다.

<26>
$$E[V] = \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} \int_0^{2\pi} \int_0^{2\pi} \sqrt{v_1^2 + v_2^2 - 2v_1v_2 \cos(\phi_1 - \phi_2)} f_{V_1, V_2, \Phi_1, \Phi_2}(v_1, v_2, \phi_1, \phi_2) d\phi_1 d\phi_2 dv_1 dv_2$$

<27> 여기서 $f_{V_1, V_2, \Phi_1, \Phi_2}(v_1, v_2, \phi_1, \phi_2)$ 는 랜덤 변수들 v_1, v_2, ϕ_1, ϕ_2 의 조인트 pdf이고, V_{min} 및 V_{max} 는 최소 이동 속도 및 최대 이동 속도이고, 기호 $E[V]$ 는 랜덤 변수 V 의 평균값이다. 노드 S의 이동 속도 v_1, v_2 와 방향 ϕ_1, ϕ_2 는 독립적이므로, 상기 수식은 아래의 수식과 같이 간략화될 수 있다.

<28>
$$E[V] = \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} \int_0^{2\pi} \int_0^{2\pi} \sqrt{v_1^2 + v_2^2 - 2v_1v_2 \cos(\phi_1 - \phi_2)} f_V(v_1) f_V(v_2) f_\Phi(\phi_1) f_\Phi(\phi_2) d\phi_1 d\phi_2 dv_1 dv_2$$

<29> 만일 ϕ_1 및 ϕ_2 가 $[0, 2\pi]$ 내에서 균일하게 분산되었다면 상기 수식은 아래의 수식과 같이 다시 쓸 수 있다.

<30>
$$E[V] = \frac{1}{\pi^2} \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} (v_1 + v_2) F_e\left(\frac{2\sqrt{v_1v_2}}{v_1 + v_2}\right) f_V(v_1) f_V(v_2) dv_1 dv_2$$

<31> 여기서, $F_e(k) = \int_0^1 \sqrt{\frac{1-k^2t^2}{1-t^2}} dt$ 는 두 종류의 타원적 적분으로 완성된다. 그러므로, 현재 노드 S는 멈추어 있고, 노드 U는 상대 속도로 이동한다고 생각할 수 있다. 노드들이 균일하게 분산되어 있고, 노드들의 이동 방향이 $[0, 2\pi]$ 상에 균일하게 분산되어 있다고 가정하면 R_{dwell} 의 평균값은 아래의 수식과 같이 주어진다.

<32>
$$R_{dwell} = \frac{E[V]L}{\pi A}$$

<33> 여기서 A 는 전송 범위 영역이고, L 은 이 영역의 페리미터(perimeter)이다. 그러므로,

<34>
$$E[T_{dwell}] = \frac{\pi A}{E[V]L}$$

<35> 따라서, 예측 간격은 $E[T_{dwell}]$ 시간 내에서 경계 지어질 수 있다.

<36> 이동성 예측

<37> Camp et al은 MANETs에 대한 이동성 모델들에 대한 포괄적인 연구를 수행하였으며, 이로부터 우리는 일부 모델들에서 어떤 노드들이 방향 변환 전에 선형으로 움직임을 발견할 수 있다. 또 다른 모델들에서는 그들이 정확하게 선형 움직임을 아니지만 노드들이 또한 세그먼트 뷰에서는 또한 선형으로 움직인다.

<38> 위치 기반 예측 : (x_{1h}, y_{1h}, z_{1h}) 및 (x_{2h}, y_{2h}, z_{2h}) 의 위치 정보를 갖는 t_{1h} 와 $t_{2h}(t_{1h} > t_{2h})$ 각각의 시간에서 특정 노드에 두 개의 최신 정보가 있음을 가정하자. 적어도 두 개의 성공적인 업데이트 기간 내에 노드는 도 3a에 도시한 바와 같이 고정된 속도를 갖고 직선으로 움직인다고 가정하면, 아래의 수식을 얻는다.

<39>
$$\begin{cases} \frac{x_{1h} - x_{2h}}{t_{1h} - t_{2h}} = \frac{x_p - x_{1h}}{t_p - t_{1h}} \\ \frac{y_{1h} - y_{2h}}{t_{1h} - t_{2h}} = \frac{y_p - y_{1h}}{t_p - t_{1h}} \\ \frac{z_{1h} - z_{2h}}{t_{1h} - t_{2h}} = \frac{z_p - z_{1h}}{t_p - t_{1h}} \end{cases}$$

<40> 그리고, 미래 시간 t_p 에서의 위치 (x_p, y_p, z_p) 는 아래의 수식과 같이 연산될 수 있다.

$$\begin{cases} x_p = x_{1h} + \frac{x_{1h} - x_{2h}}{t_{1h} - t_{2h}}(t_p - t_{1h}) \\ y_p = y_{1h} + \frac{y_{1h} - y_{2h}}{t_{1h} - t_{2h}}(t_p - t_{1h}) \\ z_p = z_{1h} + \frac{z_{1h} - z_{2h}}{t_{1h} - t_{2h}}(t_p - t_{1h}) \end{cases}$$

<42> 그러나, 전형적인 업데이트에서, 이 모델은 최신 업데이트가 이전의 업데이트와 비교하여 많은 변화들을 나타내기 때문에 사용될 수 없다.

<43> 속도 보정 예측 : 특정 노드에 대한 그것의 최신 업데이트의 속도가 (v'_x, v'_y, v'_z) 라고 하자. 노드는 도 3b에 도시한 바와 같이 업데이트 기간 내의 속도로 움직인다고 가정하면, 미래 시간 t_p 에서의 위치

(x_p, y_p, z_p) 는 아래의 수식과 같이 연산될 수 있다.

$$\begin{cases} x_p = x_{1h} + v'_x(t_p - t_{1h}) \\ y_p = y_{1h} + v'_y(t_p - t_{1h}) \\ z_p = z_{1h} + v'_z(t_p - t_{1h}) \end{cases}$$

<45> 고속 이동성 네트워크에서 이동 노드 상의 힘은 변하지 않고, 노드들은 등가속도로 움직인다고 가정할 수 있다.

<46> 등가속도 예측 : (v'_x, v'_y, v'_z) 및 (v''_x, v''_y, v''_z) 는 각각 도 3c에 도시한 바와 같이 두 개의 업데이트의 속도이다. 운동 법칙의 원리는 아래의 수식과 같다.

$$V = v + at$$

<48> 그리고

$$S = vt + \frac{1}{2}at^2 = \bar{v}t = \frac{v + V}{2}t$$

<50> 여기서 S는 변위(displacement)이고, v는 처음 속도(initial velocity)이며, a는 t 기간 동안의 가속도이다.

우리는 t 기간 이후의 최종 속도를 나타내는 V를 사용한다. 고정된 가속도를 (a_x, a_y, a_z) 로 가정하고, 상기 원칙을 X-dimension에 적용하여 아래의 수식을 얻는다.

$$\begin{cases} v'_x = v''_x + a_x(t_{1h} - t_{2h}) \\ v_x = v'_x + a_x(t_p - t_{1h}) \\ x_p - x_{1h} = \frac{(v'_x + v_x)}{2}(t_p - t_{1h}) \end{cases}$$

<52> 그리고, 예상되는 위치 x_p 를 아래의 수식과 같이 얻을 수 있다.

$$x_p = x_{1h} + \frac{2v'_x + (v'_x - v''_x)\frac{t_p - t_{1h}}{t_{1h} - t_{2h}}}{2}(t_p - t_{1h})$$

<54> Y 및 Z dimension들은 X-dimension과 같으므로, 아래의 수식을 얻는다.

$$\begin{cases} x_p = x_{1h} + \frac{2v'_x + (v'_x - v''_x)\frac{t_p - t_{1h}}{t_{1h} - t_{2h}}}{2}(t_p - t_{1h}) \\ y_p = y_{1h} + \frac{2v'_y + (v'_y - v''_y)\frac{t_p - t_{1h}}{t_{1h} - t_{2h}}}{2}(t_p - t_{1h}) \\ z_p = z_{1h} + \frac{2v'_z + (v'_z - v''_z)\frac{t_p - t_{1h}}{t_{1h} - t_{2h}}}{2}(t_p - t_{1h}) \end{cases}$$

<56> 마지막으로, 예측되는 위치를 모음으로써, 업데이트되고 모순이 없는(consistent) 인접노드(neighborhood) 로컬 뷰를 구성할 수 있다.

<57> 또한, 본 발명은 예측적이고 동기화된 솔루션을 제공하지만, 부정확한 로컬 뷰를 야기할 수 있는 또 다른 상황이 존재한다. 그것은, 노드 S가 노드 U의 최신 업데이트를 수신받지 못해서 S가 U의 존재를 간과하는 것이다. 그러나, U는 예측 시간 동안 노드 S의 인접노드(neighborhood)로 움직인다. 도 4a는 노드 S의 예측된 로컬 뷰를 나타내고, 여기에서 노드 U는 비록 그들이 S의 근처에 있을지라도 포함되지 않는다.

<58> 상기 언급된 문제를 해결하기 위하여, 본 발명은 더 작은 인접노드(neighborhood) 반경(SR)을 적용하여 S의 인접노드(neighborhood) 로컬 뷰를 재구성하는 방법을 제공한다. SR 기법을 적용함으로써, 노드 S는 도 4b에 도시한 바와 같이 더 작지만 정확한 로컬 뷰를 얻는다.

<59> 도 5에 도시한 바와 같이 두 개의 노드 S와 U에서, 노드 U는 시간 t_0 에 노드 S의 전송 범위 내에 없고, t_1 에 U' 위치로 이동한다. t_0 에서 그들의 거리는 d이고 U는 t_1 에서 S에 대하여 x의 거리를 이동한다고 가정하자. U가 S의 전송 범위로 들어갈 확률은

<60>
$$p(x, d) = \begin{cases} 0: & x < d - R_1 \\ \frac{x}{\pi}: & d - R_1 \leq x \leq d + R_1 \\ 0: & x > d + R_1, \end{cases}$$

<61> 여기서 $\phi = \arccos\left(\frac{x^2 + d^2 - R_1^2}{2xd}\right)$ 는 $R_2 < R_1$ 을 충족하는 $\angle SUU'$ 의 가장 큰 값이다.

<62> 어떤 노드가 t_1 에서 노드 S의 전송 범위로 이동할 확률은

<63>
$$p(x) = \int_{R_1}^{\infty} p(x, d) dd$$

<64> S에 대하여 어떤 상대 속도 v를 갖는 노드가 그것의 전송 범위로 이동할 확률은

<65>
$$p = \int_0^{2s} f_{|\vec{V}|}(v) p(fv) dv$$

<66> 여기서 $\vec{V}$ 는 전송한 임의의 상대속도 벡터이고, s는 어떤 노드의 최대속도이다. 다시 말하면, $\vec{V}$ 의 방향은 또한 $[0, 2\pi]$ 에 균일하게 분산되어 있고, $\vec{V}$ 의 속도와 관계없다. 또한, $|\vec{V}|$ 는 $[0, 2s]$ 에서 균일하게 분산된다.

<67> 아래의 수식과 같이 시간 t가 주어질 때 $f_{|\vec{V}|}$ 을 연산한다.

<68>
$$\begin{aligned} f_{|\vec{V}|}(t) &\approx \frac{F_{|\vec{V}|}(\delta_t) - F_{|\vec{V}|}(t)}{\delta_t} \\ &= \frac{P(t \leq |\vec{V}| \leq t + \delta_t)}{\delta_t} \\ &= \frac{\int_{(0,0)}^{(2\pi,s)} \int_{(0,0)}^{(2\pi,s)} \frac{R(\vec{V}_2, \vec{V}_1, t, t + \delta_t)}{(2\pi s)^2 \delta_t} d\vec{V}_2 d\vec{V}_1}{\delta_t} \end{aligned}$$

<69> 여기서 $F_{|\vec{V}|}(t)$ 는 분산 함수이고, δ_t 는 작은 양수이고,

<70>
$$R(\vec{V}_2, \vec{V}_1, a, b) = \begin{cases} 1: & a \leq |\vec{V}_2 - \vec{V}_1| \leq b \\ 0: & otherwise. \end{cases}$$

<71> 상기 모든 공식들을 결합하면, 어떤 노드 U가 노드 S의 전송 범위로 이동하는 것을 연산할 수 있으며, 보다 작은 인접노드(neighborhood) 범위(SR)의 예상 값은 아래의 수식에 의해 얻는다.

$$E[SR] = (1 - p)R_1$$

본 발명의 시뮬레이션 툴은 ns-2.28 및 CMU 무선 확장(extension)을 사용하고, 아래의 표 1에 나열된 것과 같은 변수들을 갖는 무선 노드 모델로서, AT&T의 Wave LAN PCMCIA 카드로 가정한다. 본 발명의 포괄적인 유효성을 증명하기 위하여, MANETs를 위해 설계된 시뮬레이팅 프로토콜에서 넓게 사용되는 선형(Random Waypoint)뿐 아니라 비선형(Gauss-Markov) 이동성 모델들에서 실험을 실행한다.

파라미터들	값
주파수	2.4 GHz
최대 전송 범위	250m
맥(MAC) 프로토콜	802.11
전파(propagation) 모델	프리 영역(free space)/ 투 레이 그라운드(two lay ground)

표 1. 이동 노드 모델에 대한 파라미터(parameter)들

인접노드(neighborhood) 추적(tracking)에서는 어떠한 노드 S가 로컬 뷰 구성을 위한 그것의 인접(neighbor) 노드의 위치들을 예측하기 위해 임의대로 선택된다.

로컬 뷰의 구성은 업데이트 간격(interval) 이내에서 발생한다. 아래의 표 2는 시뮬레이션 변수들을 표시한다. 정기적인 업데이트 하에서 속도 기반 예측을 갖는 예측되는 로컬 뷰의 샘플이 도 6에 도시되어 있고, 또한 실제 로컬 뷰 및 업데이트 정보에 기초하는 로컬 뷰의 비교를 나타낸다.

파라미터들	값
시뮬레이션 네트워크 사이즈	900×900 m ²
이동 노드 속도 범위	[0, 15]m/s
노드 갯수	50
시뮬레이션 타임	50s
정기적인 업데이트/체크 간격	2s
예측 간격	20ms
전형적인 업데이트의 참조 길이	1m

표 2. 시뮬레이션 파라미터(parameter)들

로컬 뷰의 부정확을 평가하기 위해, 본 발명은 네이버(neighbor)의 실제 위치와 인접노드(neighborhood) 뷰 내에서 그들의 위치 간의 평균 거리 차와 같은 위치 오류(PE)의 metric을 정의한다. 어떠한 노드 S에 대해 그것의 j번째 로컬 뷰에 K의 neighbor들(S 자신을 포함)이 있는 것을 제안하고, j번째 인접노드(neighborhood)에 대한 PE_j는 아래의 수식과 같이 연산될 수 있다.

$$\sqrt{\frac{1}{K} \sum_{i=1}^K [(x'_i - x_i)^2 + (y'_i - y_i)^2 + (z'_i - z_i)^2]}$$

마지막으로, W 로컬 뷰를 갖는 것을 가정하면

$$PE = \frac{1}{W} \sum_{j=1}^W PE_j$$

PE의 값이 작으면 작을수록, 인접노드(neighborhood) 로컬 뷰도 더 정확해진다.

아래의 표 3과 표 4는 우리의 시뮬레이션 내에서 Random Waypoint와 Gauss-Markov의 위치 오류 결과를

나타낸다.

레코드 타입	예측 방법	PE 값
정기적인 업데이트	업데이트 정보 기반	7.258410
	위치-기반	0.755039
	속도-보정	0.003444
	등가속도	0.261483
전형적인 업데이트	업데이트 정보 기반	9.267584
	속도-보정	0.000006
	등가속도	0.637606

표 3. random waypoint model에서의 PE

레코드 타입	예측 방법	PE 값
정기적인 업데이트	업데이트 정보 기반	7.407275
	위치-기반	2.281239
	속도-보정	0.497334
	등가속도	1.046533
전형적인 업데이트	업데이트 정보 기반	9.497269
	속도-보정	1.617758
	등가속도	2.813394

표 4. Gauss-Markov model에서의 PE

상기 시뮬레이션 결과로부터 우리는 다음과 같은 사실을 증명할 수 있다.

정기적이고 조건적인 업데이트 정보 기반 뷰는 본 발명의 추적(tracking) 방법과 비교하여 세 번 이상의 오류를 갖는데, 이것은 본 발명의 필요성 및 유효성을 증명하는 것이다.

본 발명은 매우 작은 예측 오류(특히 선형 이동성 환경에서)를 갖는다. 즉, 정밀하게 인접노드(neighborhood)를 추적(tracking)할 수 있다.

다른 예측 모델들은 다른 성능(performance)을 갖는다. 속도-보정 방법은 다른 두 방법보다 훨씬 잘 실행하고, 등가속도 모델은 위치-기반 방법보다 낫다. 또한, 이동성 모델과 업데이트 프로토콜도 우리 기법의 실행에 영향을 미친다. 다른 이동성 모델들과 업데이트 프로토콜 하에 PE값도 또한 다르다.

도면의 간단한 설명

도 1은 본 발명의 실시 예에 따른 인접노드(neighborhood) 추적(tracking) 기법을 나타내는 순서도이다.

도 2는 예측 간격에 대한 분석 모델을 나타내는 도면이다.

도 3a, 3b 및 3c는 위치 기반, 속도 보정 및 등가속도 예측 모델을 나타내는 도면이다.

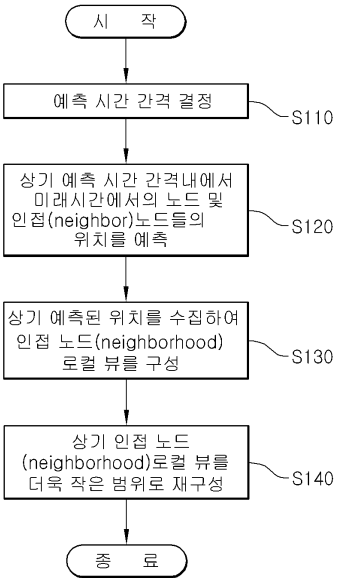
도 4는 보다 작은 인접노드(neighborhood) 범위의 함수를 나타내는 도면이다.

도 5는 보다 작은 인접노드(neighborhood) 범위의 분석 모델을 나타내는 도면이다.

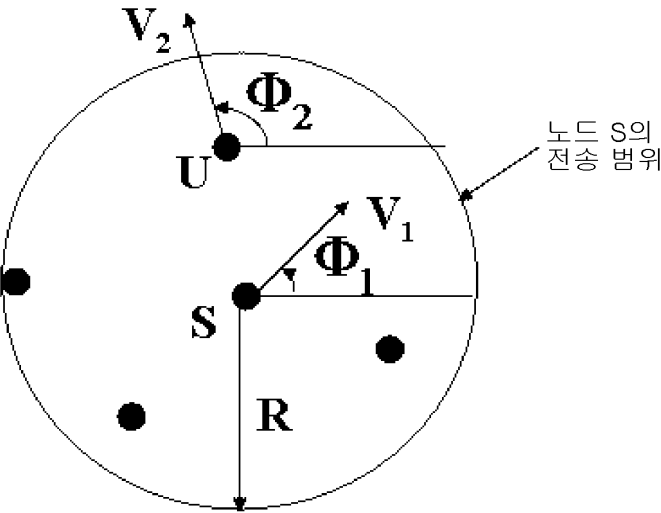
도 6a, 6b는 모든 노드들의 Z 차원 좌표들이 0으로 설정된 것을 나타내는 도면이다.

도면

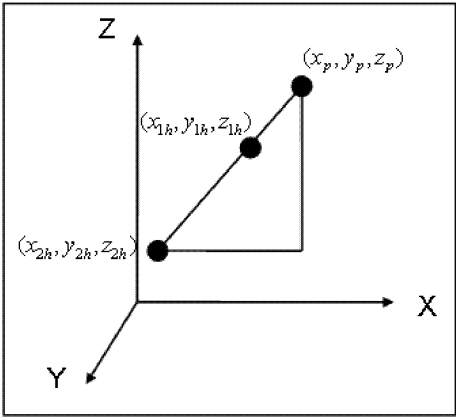
도면1



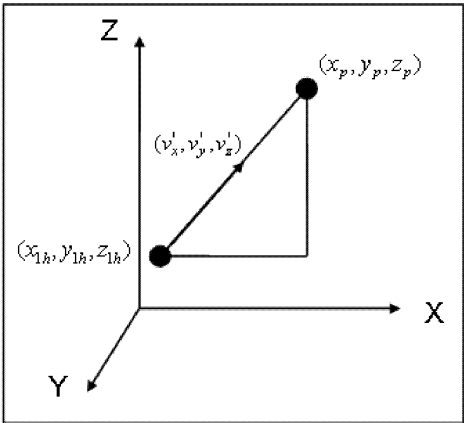
도면2



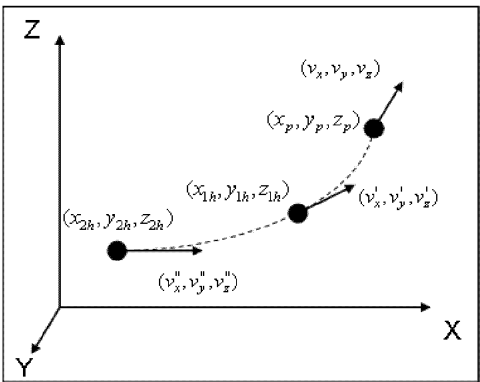
도면3a



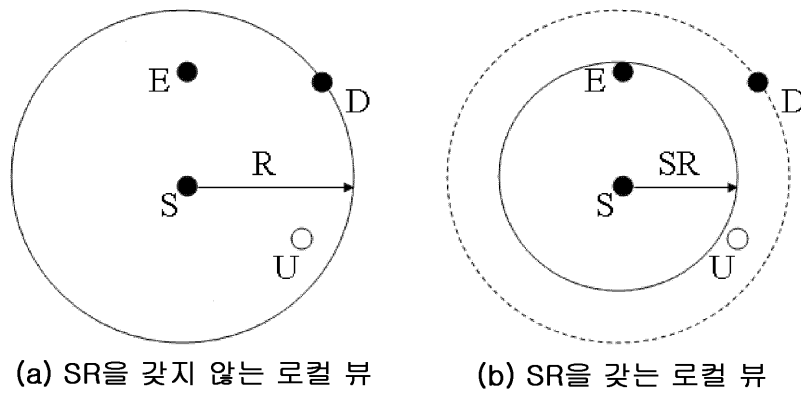
도면3b



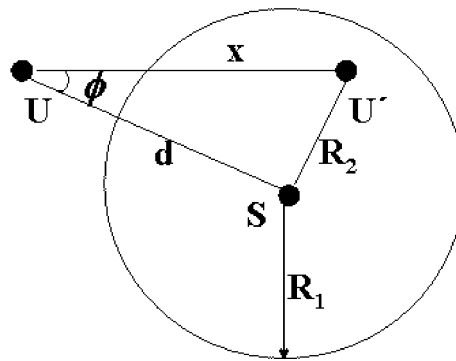
도면3c



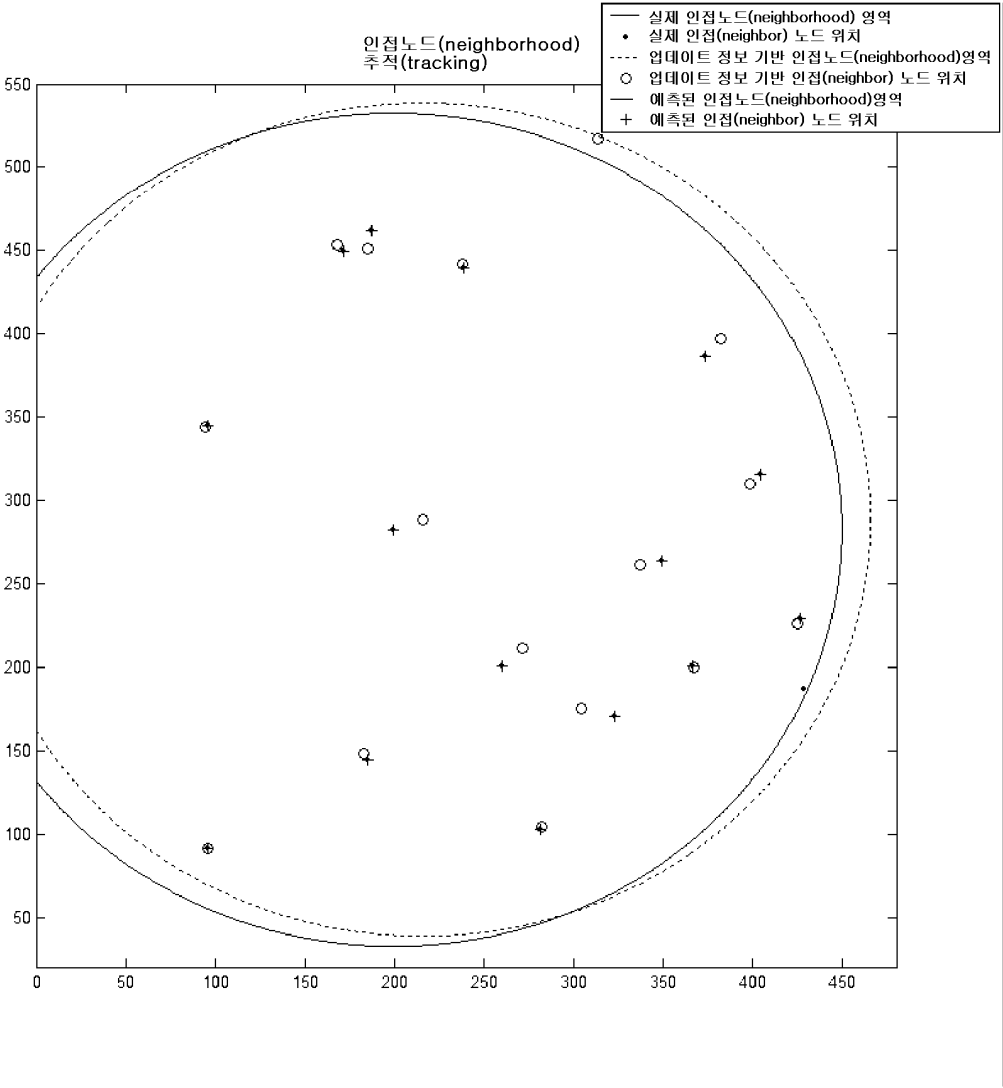
도면4



도면5



도면6a



도면6b

