

## Delay Tolerant Networks에서 속성정보 예측 모델을 이용한 상황인식 연결성 분석 기법

정래진<sup>1</sup> · 오영준<sup>2</sup> · 이강환<sup>3\*</sup>

### Context-aware Connectivity Analysis Method using Context Data Prediction Model in Delay Tolerant Networks

Rae-jin Jeong<sup>1</sup> · Young-jun Oh<sup>2</sup> · Kang-whan Lee<sup>3\*</sup>

<sup>1</sup>Interdisciplinary Program in Creative Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

<sup>2</sup>Department of Computer Science Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

<sup>3\*</sup>Department of Computer Science Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

#### 요 약

본 논문에서는 계층적 클러스터 구조에서 속도, 방향 등의 속성정보를 예측하여 연결성을 분석하는 EPCM(Efficient Prediction-based Context-awareness Matrix) 알고리즘을 제안한다. 기존 DTN에서는 제한 없는 중계노드 선정으로 인한 지연시간이 증가하고, 제한적인 저장용량 및 처리능력의 한계로 패킷손실 및 오버헤드가 발생한다. 따라서 본 논문에서는 클러스터의 속성정보를 주어진 매트릭으로부터 정의하고 예측하여 베이스스테이션과의 연결성에 따라 중계노드를 선정하는 EPCM 알고리즘을 제안한다. 제안된 알고리즘은 클러스터의 속성정보를 상황정보 매트릭스에 저장 및 분석하고, 적응적 보정가중치의 오차보정으로 클러스터의 속성정보를 예측한다. 또한 예측된 속성정보에 따른 연결성으로 베이스스테이션을 향하여 이동하는 중계노드를 선정하게 된다. 주어진 환경에 따른 모의실험에서 예측된 노드의 속성정보를 통한 연결성으로 중계노드를 선정함으로써 전송능력을 비교 분석한 결과 높은 패킷 전송률을 보여주었다.

#### ABSTRACT

In this paper, we propose EPCM(Efficient Prediction-based Context-awareness Matrix) algorithm analyzing connectivity by predicting cluster's context data such as velocity and direction. In the existing DTN, unrestricted relay node selection causes an increase of delay and packet loss. The overhead is occurred by limited storage and capability. Therefore, we propose the EPCM algorithm analyzing predicted context data using context matrix and adaptive revision weight, and selecting relay node by considering connectivity between cluster and base station. The proposed algorithm saves context data to the context matrix and analyzes context according to variation and predicts context data after revision from adaptive revision weight. From the simulation results, the EPCM algorithm provides the high packet delivery ratio by selecting relay node according to predicted context data matrix.

**키워드** : DTN, 상황인식, 연결성, 매트릭스

**Key word** : Delay Tolerant Networks, Context-awareness, Connectivity, Matrix

Received 17 December 2014, Revised 19 January 2015, Accepted 02 February 2015

\* Corresponding Author Kang-Whan Lee(E-mail:kwlee@koreatech.ac.kr, Tel:+82-41-560-1356)

Department of Computer Science Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.4.1009>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

이동노드로 구성된 무선 네트워크 환경에서 유동적인 연결성으로 인한 Opportunistic Communication은 사회적 관계성, 이동성 패턴 분석 등의 연구와 융합되어 많은 연구가 이루어지고 있다. Opportunistic Communication는 이동노드들이 별도의 연결을 확립하지 않고 서로에게 근접하면 통신을 확립하는 특성으로 바다, 사막과 같이 드넓은 지역 혹은 도시 내부에서 교통수단을 활용한 통신을 목적으로 연구되고 있다[1, 2]. 또한 스마트폰, 태블릿 PC 등 와이파이, 블루투스, 각종 센서 등이 내장된 모바일 기기가 널리 보급되면서 기기의 근접 여부를 통해 소유자의 관계성을 기반으로 하여 소유한 사물 간의 관계를 설정하는 연구로 확대되고 있다[3]

DTN(Delay Tolerant Networks)은 부분적인 연결성을 이용하여 데이터 패킷이 이동하는 특성으로 다양한 연구가 제시되고 있다. 특히 노드의 상황정보 혹은 사회적 관계를 활용하여 중계노드를 선정하는 연구가 전개되고 있다. DTN의 대표적인 중계노드 선정 방법은 상황 기반 방식과 사회기반 방식으로 구분할 수 있다. 상황 기반 방법은 연결 횟수, 시간 등으로 중계노드를 선정하는 방법이고, 사회 기반 방법은 비슷한 사회성을 지닌 노드를 커뮤니티로 구성하여 중계노드를 선정하는 방법이다[1, 2].

만일 중계노드 선정 과정에서 목적 노드와는 다른 속성을 가진 중계노드가 선정되었을 경우 추가적인 중계로 인한 지연시간이 증가하여 통신이 원만히 이루어지지 않게 된다. 또한 중계노드의 제한적인 저장용량 및 처리능력의 한계로 인한 패킷손실 및 오버헤드가 발생하여 네트워크의 성능을 저하시키는 문제점이 발생한다. 따라서 효율적인 중계노드를 선정하여 데이터 패킷을 전송하는 방법이 필요하다.

본 논문에서는 예측된 클러스터의 속성정보를 이용한 상황인식 중계노드 선정 기법인 EPCM(Efficient Prediction-based Context-awareness Matrix)알고리즘을 제안한다. 제안하는 EPCM 알고리즘은 계층적 클러스터 구조에서 노드의 예측 속도와 방향에 따라 베이스스테이션과의 연결성을 고려하여 중계노드를 선택하는 기법이다. 주어진 모의실험환경에서 노드의 속도와 방향에 따른 이동성을 예측함으로써 효율적인 패킷 전송

를 보여주었다.

## II. 관련연구

데이터 전송을 중계노드의 이동성에 의존하는 DTN의 특성에 따라 이동노드의 속성정보를 기반으로 하여 효율적인 중계노드를 선택하는 연구가 진행되어 왔다 [4-8]. 대표적으로 노드들의 연결 횟수와 시간에 따른 연결가능성으로 중계노드를 선택하는 알고리즘인 PROPHET(Probabilistic ROuting Protocol using History of Encounter and Transitivity)은 노드들이 연결된 경우 두 노드에 대한 연결가능성이 증가, 연결되지 않은 경우 시간에 따라 연결가능성이 감소하게 된다. 또한 제 3의 노드를 거쳐 연결되는 경우를 고려하여 연결가능성을 계산한다. 소스노드는 연결이 확립되어 있는 노드 중에 목적노드와 연결가능성이 가장 높은 노드를 선택하여 데이터 패킷을 전달한다[4].

노드의 사회적인 관계성을 중계노드 선정에 고려한 BUBBLE Rap 알고리즘은 노드들을 커뮤니티로 구분한다[5]. 데이터 패킷은 커뮤니티에서 Betweenness Centrality가 높은 노드들의 중계를 통해 목적노드가 소속된 커뮤니티를 향해 이동하여 전송된다.

BUBBLE Rap의 커뮤니티와 같이 노드들의 속도, 방향 등의 속성정보들을 활용한 계층적 클러스터 구조를 생성하고 관리하는 DDV-hop 알고리즘이 제시되었다 [9]. DDV(Dynamic Direction Vector)-hop 알고리즘은 베이스스테이션을 기준으로 전체 네트워크 영역이 4개의 구역으로 나누어지며, 각 구역에 위치한 노드들 중에서 각각 속도와 방향이 다른 클러스터 헤드노드를 선정한다. 이 후 다른 노드들은 자신의 속도와 방향의 유사한 정도에 따라 클러스터 헤드노드를 선택하여 클러스터가 구성된다.

## III. 본 론

본 논문에서는 클러스터의 이동을 예측하고 베이스스테이션과의 연결성을 고려하여 중계노드를 선정하는 EPCM 알고리즘을 제안하였다. 기존의 중계노드 선정 기법은 다른 속성을 가진 중계노드 선정으로 지연시간

이 증가하고 네트워크 자원이 불필요하게 소비되는 결과가 나타난다. 따라서 본 논문에서는 DDV-hop 알고리즘으로 구성된 속도와 방향에 따른 클러스터 구조에서 클러스터 헤드노드의 속성정보를 저장 및 분석하여 노드의 속성정보를 예측한다. 그림 1과 같이 이웃 클러스터의 예측된 속성정보를 활용하여 베이스스테이션으로 이동함에 따른 연결성을 분석하는 방법을 표현하고 있다.

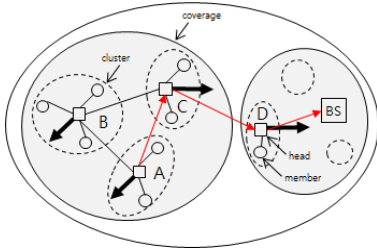


그림 1. 속성정보에 따른 중계노드 선택

Fig. 1 The Selection of relay node according to context data

### 3.1. 상황정보 매트릭스

제안하는 EPCM 알고리즘은 DDV-hop 알고리즘으로부터 얻은 각 클러스터 헤드노드의 속도와 방향으로 클러스터의 이동성을 예측한다. 그림 2와 같이 연속적인 클러스터의 속성정보는 타임 슬롯 테이블의 매트릭스 형태로 저장되며, 생성시간에 따라 누적되어 정렬된다. 이러한 타임 슬롯의 집합 형태로 속성정보들을 구조화한 것을 상황정보 매트릭스라 한다.

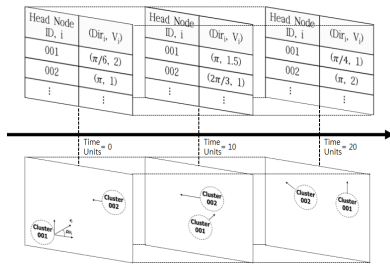


그림 2. 상황정보 매트릭스에서 노드의 상황정보

Fig. 2 The Node's context data in Context Matrix

상황정보 매트릭스는 시간  $t, \{t_0, t_1, \dots, t_n\}$ 에 따라 생성된 타임 슬롯의 집합  $U$ 로 정의하며, 다음 수식 1과 같이 표현할 수 있다.

$$U = \{ts(t_0), ts(t_1), \dots, ts(t_{n-1}), ts(t_n)\} \quad (1)$$

여기서,  $ts(t_0)$ 는 시간  $t_0$ 에 생성된 타임 슬롯을 의미한다.

### 3.2. 상황정보 변화량 분석

근접한 타임 슬롯에 저장되어 있는 속성정보의 차를 분석하여 해당 시간 간격에 클러스터 상황정보의 변화를 파악할 수 있다. 따라서 EPCM 알고리즘은 두 타임 슬롯에 저장된 값의 차이를 통해 속성정보의 변화 정도를 계산한다. 수식 2와 3은 속도와 방향의 변화량의 차이를 나타내고 있다.

$$\Delta v_i^{u,w} = \frac{|v_i^w - v_i^u|}{v_i^u} \quad (2)$$

$$\Delta Dir_i^{u,w} = \frac{|Dir_i^w - Dir_i^u|}{Dir_i^u} \quad (3)$$

여기서,  $v_i^u, Dir_i^u$ 는 타임 슬롯  $u$ 에 저장되어 있는 클러스터 헤드노드  $i$ 의 속도와 방향을 의미하며, 마찬가지로  $v_i^w, Dir_i^w$ 는 타임 슬롯  $w$ 에 저장되어 있는 클러스터 헤드노드  $i$ 의 속도와 방향이다.  $\Delta v_i^{u,w}$ 와  $\Delta Dir_i^{u,w}$ 는 각각 클러스터 헤드노드  $i$ 의 속도와 방향 변화량의 비를 나타낸다.

수식 4를 통해 속도와 방향의 변화 정도를 통합적으로 나타낼 수 있고, 속도와 방향에 대한 상황정보 가중치  $\omega_{cd}(v), \omega_{cd}(Dir)$ 로 속도와 방향 중 특정 속성정보의 변화를 중시하여 이동성을 분석할 수 있다.

$$\Delta \alpha_i(v, Dir) = \frac{1}{N_{cd}} (\omega_{cd}(v) \Delta v_i^{u,w} + \omega_{cd}(Dir) \Delta Dir_i^{u,w}) \quad (4)$$

여기서,  $\Delta \alpha_i(v, Dir)$ 는 타임 슬롯  $u$ 와  $w$ 의 생성 시간 동안에 클러스터 헤드노드  $i$ 의 상황정보 변화 정도를 나타낸 상황정보(context data) 변화량이다.  $\omega_{cd}(v)$ 와  $\omega_{cd}(Dir)$ 은 상황정보 가중치이며 두 가중치의 합은 1이며,  $N_{cd}$ 는 상황 정보의 종류의 수를 의미한다.

#### 3.2.1. 타임 슬롯 생성 주기

특정 클러스터가 일정한 속도와 방향으로 이동하는 경우 생성되는 타임 슬롯은 같은 속성정보 값을 연속

해서 저장하게 된다. 이는 중복 데이터의 량이 증가하여 속성정보의 관리 및 예측연산의 결과에도 오차가 증가할 수 있다. 따라서 EPCM 알고리즘은 그림 3과 같이 상황정보 변화량을 통해 저장 주기를 가변적으로 조정한다.

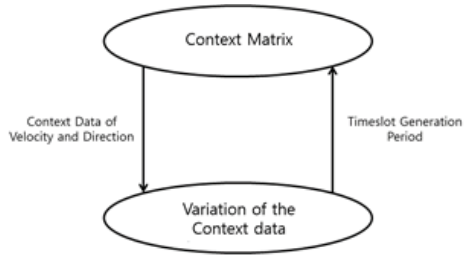


그림 3. 상황정보 변화량에 따른 타임 슬롯 생성주기 조정  
Fig. 3 The adjustment of time slot generation period according to variation of context data

타임 슬롯 생성주기는 상황정보의 변화량에 따라 가변적으로 변화하며 다음 수식 5와 같이 표현할 수 있다.

$$T_i = \max[T_{\min}, T_{\max} \times \gamma^{\left| \frac{\Delta cd_i(v, Dir)}{tr_w - tr_u} \right|}] \quad (5)$$

여기서,  $T_i$ 는 갱신된 저장주기이며,  $T_{\min}$ 과  $T_{\max}$ 는 각각 최소 저장 주기, 최대 저장 주기이다.  $tr_w$ 와  $tr_u$ 는 타임 슬롯  $u$ 와  $w$ 의 생성시간이다.  $\gamma$ 는 0에서 1사이의 값을 가지는 상수이다.

### 3.2.2. 타임 슬롯 구간

예측과정에서 정확도를 향상시키기 위해 최신의 타임 슬롯부터 속성 정보의 변화 추이가 같은 타임 슬롯들을 구간화한다. 따라서 구간으로 설정된 타임 슬롯안의 속성정보만을 예측연산에 활용함으로써 예측에서 불필요한 데이터를 제외한다. 수식 6을 활용하여 구간을 나누는 기준인 임계 값  $\alpha$ 을 표현할 수 있다.

$$\alpha = \frac{1}{n(U)} \sum \Delta cd_i(v, Dir) \quad (6)$$

여기서,  $n(U)$ 는 수식 1에서 정의한 모든 타임 슬롯으로 구성된 집합  $U$ 에서 속하는 타임 슬롯의 수이고,  $\Delta cd_i(v, Dir)$ 는 상황정보의 변화량으로 나타낸다.

구분된 각각의 구간들은 유지구간 (MI : Maintenance Interval), 상위구간 (HI : High rank Interval), 하위구간 (LI : Low rank Interval)으로 설정되며, 각 상황에 따라 구간을 수식 7, 8, 9과 같이 표현 할 수 있다.

$$MI = \{u, w | \Delta cd_i(v, Dir) = 0\} \quad (7)$$

$$HI = \{u, w | \Delta cd_i(v, Dir) > \alpha\} \quad (8)$$

$$LI = \{u, w | \Delta cd_i(v, Dir) < \alpha\} \quad (9)$$

### 3.3. 적응적 보정 가중치

클러스터의 이동성은 실시간으로 다양하게 변화하기 때문에 예측의 오차가 발생할 수 있다. 그림 4와 같이 실제 속성정보 값과 이전 예측한 속성정보의 값의 오차 크기에 따른 적응적 보정 가중치를 활용한 피드백 과정을 거쳐 적응적으로 예측 오차를 보정한다.

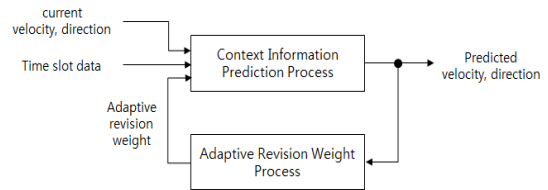


그림 4. 적응적 보정 가중치의 적용  
Fig. 4 The application of adaptive revision weight

속도와 방향에 대한 적응적 보정 가중치  $\omega_i(v)$ 와  $\omega_i(Dir)$ 은 다음 수식 10과 11와 같이 표현할 수 있다.

$$\omega_i(v) = \omega_i^{pre}(v) \times \left(1 - \left| \frac{PV_i(v) - RV_i(v)}{PV_i(v) + RV_i(v)} \right| \right) \quad (10)$$

$$\omega_i(Dir) = \omega_i^{pre}(Dir) \times \left(1 - \left| \frac{PV_i(Dir) - RV_i(Dir)}{PV_i(Dir) + RV_i(Dir)} \right| \right) \quad (11)$$

여기서,  $\omega_i^{pre}(v)$ 와  $\omega_i^{pre}(Dir)$ 은 초기 또는 과거 예측 연산에 적용했던 적응적 보정가중치이다.  $PV_i(v)$ 와  $PV_i(Dir)$ 은 예측한 속성정보 값  $PV$ (prediction value)이며,  $RV_i(v)$ 와  $RV_i(Dir)$ 는 예측된 실제 값  $RV$ (real value)을 의미한다.

### 3.4. 이동성 예측

클러스터의 속도와 방향 변화를 알아보기 위하여, 속도와 방향에 대한 평균 변화량  $v_i^{TR}$ ,  $Dir_i^{TR}$ 은 수식 12와

13과 같이 표현 할 수 있다.

$$v_i^{TG} = \frac{1}{n(TG)} \sum_{TG \in ts} (v_i^{ts} - v_i^{ts-1}) \quad (12)$$

$$Dir_i^{TG} = \frac{1}{n(TG)} \sum_{TG \in ts} (Dir_i^{ts} - Dir_i^{ts-1}) \quad (13)$$

여기서,  $TG$ 는 구간으로 정해진 타임 슬롯들의 집합이며,  $TG$ 에 속한 타임 슬롯에 대하여,  $v_i^{ts} - v_i^{ts-1}$ 는 클러스터  $i$ 의 속도차를 의미하며,  $Dir_i^{ts} - Dir_i^{ts-1}$ 는 방향 차를 나타낸다.

클러스터 헤드노드의 현재 속도와 방향의 이후 변화를  $f(v_i, Dir_i)$ 로 예측하는 방법은 다음 수식 14와 같이 표현할 수 있다.

$$f(v_i, Dir_i) = (v_i + w_i(v)v_i^{TG}, Dir_i + w_i(Dir)Dir_i^{TG}) \quad (14)$$

여기서,  $v_i, Dir_i$ 는 클러스터 헤드노드  $i$ 의 속도와 방향을 의미하며.  $w_i(v), w_i(Dir)$ 은 0에서 1까지의 값을 가지는 속도와 방향에 대한 적응적 보정 가중치이다.

### 3.5. 연결성 산출

제한하는 EPCM 알고리즘은 클러스터 헤드노드들의 예측된 속성정보를 통하여, 베이스스테이션에 대한 지향성 및 속도에 따라 연결성을 산출한다.

클러스터의 방향이 베이스스테이션을 향하는지 알아보기 위해 네트워크 중앙을 지시하는 방향과 클러스터 헤드노드의 방향 차를 다음 수식 15와 같이 나타낼 수 있다.

$$Dir_{BS,i} = \left| \left( \frac{5\pi}{4} + \frac{(quad-1)\pi}{2} \right) - Dir_i \right| \quad (15)$$

여기서,  $quad$ 는 베이스스테이션을 기준으로 나누어진 각 분면을 의미하는 상수로 1,2,3,4 중 하나의 값을 가지며,  $Dir_i$ 는 클러스터 헤드노드  $i$ 의 방향을 의미한다.

이웃한 클러스터 헤드노드 중에 데이터를 중계하기 적합한 중계노드를 알아보기 위한 연결성  $CN_{BS,i}$ 은 수식 16과 같이 표현할 수 있다.

$$CN_{BS,i} = \begin{cases} v_i (1 - \sin(Dir_{BS,i})), & (Dir_{BS,i} < \frac{\Pi}{2}) \\ 0, & otherwise \end{cases} \quad (16)$$

여기서,  $NH$ 는 클러스터 헤드 노드  $i$ 와 연결이 확립되어 있는 이웃 클러스터 헤드노드들의 집합을 나타낸다.

그림 5은 표 1의 속성정보에 따라 클러스터 헤드노드 2가 베이스스테이션에게 데이터를 전달하는 상황을 표현할 수 있다.

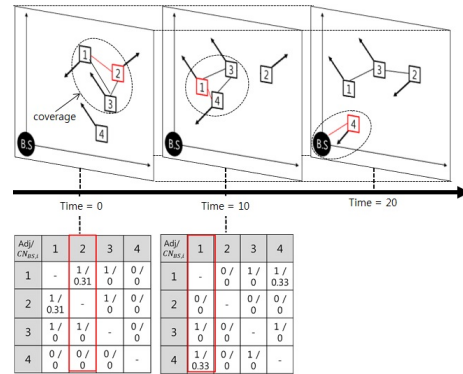


그림 5. 예측된 이동성을 통한 데이터 전송 예시

Fig. 5 The Example of Transmission according to predicted mobility

표 1. 예측된 클러스터 헤드노드들의 속도와 방향

Table. 1 The operation procedure of predicted context data

cluster head ID, $i$	1	2	3	4
Time	0	0	0	0
$v_i$	1	1	1.2	1.2
$Dir_i$	$\frac{5\Pi}{4}$	$\frac{\Pi}{4}$	$\frac{3\Pi}{4}$	$\frac{3\Pi}{4}$

cluster head ID, $i$	1	2	3	4
Time	10	10	10	10
$v_i$	1.2	1	1.2	1.2
$Dir_i$	$\frac{3\Pi}{4}$	$\frac{\Pi}{4}$	$\frac{3\Pi}{4}$	$\frac{5\Pi}{4}$

cluster head ID, $i$	1	2	3	4
Time	20	20	20	20
$v_i$	1.2	1	1.2	0.8
$Dir_i$	$\frac{3\Pi}{4}$	$\frac{5\Pi}{4}$	$\frac{3\Pi}{4}$	$\frac{5\Pi}{4}$

cluster head ID는 클러스터를 대표하는 헤드노드의 ID를 의미하며, Time은 시간을 나타낸다.  $v_i$ 와  $Dir_i$ 는 클러스터 헤드노드  $i$ 의 속도와 방향을 표현한다.

연결성을 통한 포워딩 산출 과정은 표 2의 의사코드로 나타낼 수 있다.

표 2. 중계노드 선정과정 의사코드

Table. 2 The pseudo code of select forwarding path

```

procedure : select relay node
input : source node CHi, BaseStation BS,
        set of CHi's neighbor NHi
output : relay node
begin
    initialize CHi as relay node
    for !relay node is connected to BS do
        for all NHi do
            calculate NHi's connectivity
        end
        select node of maximum connectivity
        send data to node of maximum connectivity
        set node of maximum connectivity as relay node
    end
output relay node
end
    
```

#### IV. 실험 및 분석

EPCM 알고리즘에서 예측된 속도와 방향등의 속성 정보를 통해 연결성을 도출하고, 중계노드 선정에 적용하는 방식의 효율성을 검증하고자 PROPHET 알고리즘과 패킷 전송률을 비교하였다. PROPHET 알고리즘은 두 중계노드의 접촉 여부에 따라 연결가능성을 판단하여 중계노드를 선택한다.

본 논문에서는 시간에 따른 패킷 전송률을 아래의 수식 17과 같이 표현할 수 있다.

$$PDR(t) = \frac{flow_{BS}(t)}{\sum_{i \in CH} (flow_i(t) + \sum_{j \in CM_i} flow_j(t))} \quad (17)$$

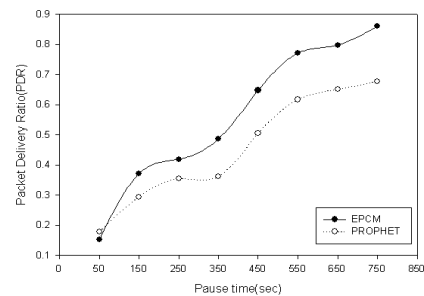
여기서,  $flow_{BS}(t)$ 는 베이스스테이션의 흐름양,  $flow_i(t)$ 는 클러스터 헤드노드  $i$ 가 받는 이웃 클러스터 헤드노드부터의 흐름양,  $flow_j(t)$ 는 클러스터에 속한 클러스터 멤버노드에 대한 흐름양이다.

주어진 실험환경은 1000 × 1000(m), 2000 × 2000(m) 크기의 네트워크 영역에 배치된 노드들에 대해 DDV-hop 알고리즘에 의하여 속도와 방향에 의한 클러스터가 구성되어 있음을 가정한다. 각 16개의 클러스터 헤드노드는 최대속도 5m/s 또는 10m/s으로 이동하여 네트워크 영역 가운데에 위치한 베이스스테이션으로 패킷을 전송하는 과정에서 속도와 네트워크 영역에 따른 패킷 전송률을 표 3과 같이 모의 실험하였다.

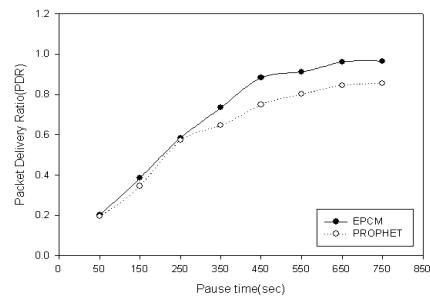
표 3. 모의 실험 환경

Table. 3 Simulation Environment

Parameter	value
Number of cluster head node	16 EA
Velocity of node	0~5 m/s, 0~10 m/s
Acceleration of node	0~5 m/s <sup>2</sup>
Network area	1000×1000(m), 2000×2000(m)
Pause time	50, 100, 250, 350, 450, 550, 650, 750 (sec)
Queue size	100 (messages)
Hop count	3,



(a)



(b)

그림 6. EPCM 알고리즘과 PROPHET 알고리즘의 최대속도에 따른 패킷전송률 비교 (a) 5m/s (b) 10m/s

Fig. 6 Comparison of packet delivery ratio about maximum velocity (a) 5m/s (b) 10m/s

그림 6은 EPCM알고리즘과 PROPHET알고리즘에 대해서 노드의 최대 속도에 따른 패킷 전송률을 비교한 결과이다.

그림 6에서 보는바와 같이 최대 속도가 증가한 경우 패킷 전송률이 증가함을 보아 중계노드의 이동성으로 데이터를 전달하는 DTN의 특성에 따라 이동성이 중계노드 선정에 중요한 요소임을 알 수 있다. 그림 7은 네트워크 영역 크기에 따른 EPCM알고리즘과 PROPHET 알고리즘의 패킷 전송률을 비교한 결과이다.

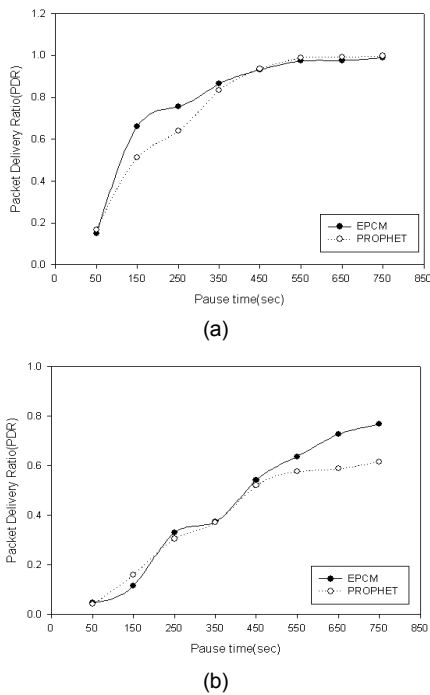


그림 7. EPCM알고리즘과 PROPHET알고리즘의 네트워크 영역 크기에 따른 패킷 전송률 비교 (a) 1000×1000(m) (b) 2000×2000(m)

Fig. 7 Comparison of packet delivery ratio about network area size (a) 1000×1000(m) (b) 2000×2000(m)

그림 7에서 보는바와 같이 1000×1000(m)환경에서 두 알고리즘은 전송한 패킷 대부분을 베이스스테이션에 전달하였지만, 네트워크 영역이 2000×2000(m)으로 확장됨에 따라 EPCM알고리즘이 더 많은 패킷을 전달하는 결과를 보여주고 있다. 따라서 EPCM알고리즘을 통해 선택되는 중계노드들이 패킷을 중앙으로 운반하여 상대적으로 베이스스테이션으로 전송할 수 있는 기

회가 많았음을 알 수 있다. 위의 결과에 따라 속도뿐만 아니라 방향도 중계노드 선정에서 큰 영향을 주고 있음을 알 수 있다.

## V. 결 론

본 논문은 클러스터의 이동 속성정보를 예측하여 베이스스테이션에 대한 지향성 및 속도에 따른 연결성을 분석한 EPCM 알고리즘을 제안하였다. EPCM 알고리즘은 속도와 방향의 유사성에 따른 계층적 클러스터 구조에서 각 클러스터 헤드노드들의 속성정보를 저장 분석하여 속도와 방향을 예측한다. 또한 예측된 속성정보를 활용하여 해당 클러스터와 베이스스테이션간의 연결성을 산출하고, 연결성이 가장 높은 클러스터 헤드노드에게 선택적으로 데이터 패킷을 전달하는 방법을 제안하였다.

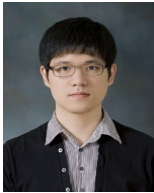
따라서 최대속도와 네트워크 영역에 따른 패킷 전송률을 비교 분석한 결과 속도와 방향과 같은 속성정보를 활용한 EPCM 알고리즘이 PROPHET 알고리즘보다 더 좋은 성능을 보임을 알 수 있었다. 단, 이동성의 변화가 빈번하고 복잡해진다면 이동성 예측 정확도가 낮아져 중계노드 선정과정에서 악영향을 줄 가능성이 있다. 향후 속도와 방향 외에 다른 속성정보를 고려하여 연결성을 세밀하게 분석하는 연구가 필요할 것이다. 더불어 CAR, HMSCAR 등의 상황인식 라우팅 프로토콜과의 비교분석을 통해 DTN환경에서 더 효율적인 전송에 관한 연구가 필요하다.

## 감사의 글

본 연구는 미래창조과학부 및 정보통신산업진흥원의 산학협력 특성화 지원사업(NIPA-2014-H0808-14-1007)과 교육부 한국연구재단의 창의 지역인력양성사업(2014H1C1A1066391) 및 한국기술교육대학교 교육연구진흥비 프로그램의 지원의 연구결과로 수행되었음

## REFERENCES

- [1] M. R. Schurgot, C. Comaniciu and K. Jaffres-Runser, "Beyond Traditional DTN Routing : Social Networks for Opportunistic Communication," *Communications Magazine IEEE*, vol. 50, no. 7, pp. 155-162, July. 2012.
- [2] L. Pelusi, A. Passarella and M. Conti, "Opportunistic Networking : Data Forwarding in Disconnected Mobile Ad Hoc Networks," *Communications Magazine IEEE*, vol. 44, no. 11, pp. 134-141, Nov. 2006.
- [3] H. Zargari Asl, A. Iera, L. Atzori and G. Morabito, "How often social objects meet each other? Analysis of the properties of a social network of IoT devices based on real data," *Global Communications Conference (GLOBECOM) 2013 IEEE*, Atlanta: GA, pp. 2804-2809, 2013.
- [4] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Service Assurance with Partial and Intermittent Resources*, Springer Berlin Heidelberg, pp 239-254, 2004.
- [5] P. Hui, J. Crowcroft and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay Tolerant Networks," *Mobile Computing IEEE Transactions*, vol. 10, no. 11, pp. 1576-1589, Nov. 2011.
- [6] M. Musolesi and C. Mascolo, "CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks," *Mobile Computing IEEE Transactions*, vol. 8, no. 2, pp. 1536-1233, Feb. 2009.
- [7] M. B. Shah, S. N. Merchant, and U. B. Desai, "Human-Mobility-Based Sensor Context-Aware Routing Protocol for Delay-Tolerant Data Gathering in Multi-Sink Cell-Phone-Based Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1-19, July. 2012.
- [8] A. Petz, A. Hennessy, B. Walker C. Fok and C. Julien, "An Architecture for Context-Aware Adaptation of Routing in Delay-Tolerant Networks," *Proceedings of the 4th Extreme Conference on Communication (ExtremeCom 2012)*, 2012.
- [9] Y. J. Oh, K. W. Lee, "Energy conserving routing algorithm based on the direction for Mobile Ad-hoc network," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 17, no. 11, pp. 2699-2707, Nov. 2013.



**정래진(Rae-Jin Jeong)**

2014년 한국기술교육대학교 컴퓨터공학부 공학사  
 2014년~ 한국기술교육대학교 창의융합공학협동과정 ICT융합전공 석사과정  
 ※관심분야 : DTN, Ad-hoc network, Ubiquitous computing



**오영준(Young-Jun Oh)**

2009년 나사렛대학교 정보통신학과 이학사  
 2011년 한국기술교육대학교 전기전자공학과 공학 석사  
 2011년~ 한국기술교육대학교 컴퓨터공학과 박사과정  
 ※관심분야 : Ubiquitous computing, WSN, Ad-hoc network, 차세대이동통신기술, Wireless SoC



**이강환(Kang-Whan Lee)**

1983년 한양대학교 전자공학과 학사  
 1989년 중앙대학교 전자공학 석사  
 1989년 한국전자통신연구원 선임연구원  
 2002년 중앙대학교 전자공학 박사  
 2004년 특허청 서기관  
 2005년~ 한국기술교육대학교 컴퓨터공학부 교수  
 ※관심분야 : WSN, Ad-hoc network, 차세대이동통신기술, Wireless SoC