

# DTN 環境を想定したトポロジ変化に強いメッセージルーティング

落合 秀也<sup>†1</sup> 江崎 浩<sup>†1</sup>

Delay (or Disruption) Tolerant Networks(DTN) の技術は、無線に代表される通信回線が不安定なネットワークで効率よくメッセージを配送する技術として有望視されている。ただ、DTN の想定する環境ではネットワークが物理的に切断されるなどのため、全体で同期を取ることが難しく、従来のルーティング方式では期待通りにメッセージを配送することができない。本研究では、ネットワーク全体で同期を取ることなく近隣ノードとの相対的な関係だけでメッセージ配送を行う Potential-Based Routing(PBR) を DTN 環境に適用し、トポロジ変化に強いルーティング手法、Topology Change Tolerant Routing(TCTR) を提案する。本研究では、プロトタイプシステムおよび TCTR シミュレータを開発した。プロトタイプシステムを使って TCTR が実装可能であることを確認すると共に、シミュレータを使い 100 台規模のノードが行動する状況において、メッセージの配送時間、転送総量、メッセージプールサイズの評価を行った。

## Topology Change Tolerant Routing for Delay Tolerant Networks

HIDEYA OCHIAI<sup>†1</sup> and HIROSHI ESAKI<sup>†1</sup>

Delay (or Disruption) Tolerant Networks (DTN) are promised as an efficient message delivery scheme in physically unstable networks like wireless networks. However, because datalinks could be physically disrupted in DTN environment, global synchronization in the network is absolutely difficult, which indicates the traditional routing schemes cannot work appropriately. We propose Topology Change Tolerant Routing (TCTR), which do not need global synchronization in the network for message delivery. In fact, TCTR is a instance of Potential-Based Routing(PBR) which selects the next hop of messages only using the relative information with its neighbor nodes. We have developed a prototype system and TCTR simulator. We have confirmed that the implementation of TCTR is feasible. We have also evaluated TCTR in terms of message delivery time, transmission cost and message pool size with the simulator.

### 1. はじめに

MANET, 無線センサネットワーク, 惑星間通信や潜水艦間通信など, 接続関係が不安定な環境下でのメッセージ配送に Delay (or Disruption) Tolerant Networks(DTN) の技術を応用することが有望視されている。IP ネットワークの場合, TCP などエンドツーエンドのセッションを張るには安定したネットワーク特性が必要であり, 先に述べた環境への適用は困難であった。一方, エンドツーエンドでセッションを張ることを想定しない DTN は, このような環境下でもメッセージを効率的に配送できると言われている。

本研究の DTN 環境は, ノード(ルータ)間の物理的な通信回線が, 接続状態と切断状態を繰り返す環境を意味することとする。DTN 環境においては, 物理

的なネットワーク・トポロジが絶えず変化しており, “現在”の全体像を把握することは, 困難である。これはネットワーク全体でコントロール信号等による同期を取ることが困難であることと等価である。ここからネットワーク全体で同期を取ることによって実現されていた従来型の経路制御プロトコル(e.g., RIP<sup>14</sup>, OSPF<sup>16</sup>, MPLS<sup>18</sup>, AODV<sup>17</sup>) が, DTN 環境には不向きであることが理解できる。

本研究は, 上記の考察を踏まえ, 物理的なトポロジの変化により, ネットワーク全体の様子が把握できない環境下でも, 近隣ノードとの関係を知るだけで, メッセージ配送を行う Topology Change Tolerant Routing(TCTR) を提案する。本研究の提案する TCTR は, Potential-Based Routing(PBR) に分類される経路制御手法で, 近隣ノードとのポテンシャルと呼ばれるスカラー値を比較し, ポテンシャル値の低い方向へメッセージを転送することで, 宛先ノードまでメッセージを配送する形態を取っている。TCTR では, ポテン

<sup>†1</sup> 東京大学

The University of Tokyo

シャルを自律分散的に生成し、ネットワーク全体のトポロジとは独立にメッセージ転送を可能にすることで、トポロジ変化に強いルーティングを実現する。

一般に DTN におけるメッセージは、Application Protocol Data Units (APDU)<sup>6)</sup> であり、IP パケットのような 1 キロバイト規模の小さなデータ単位とは限らない。数メガバイトや数ギガバイトのメッセージも対象となり、ノード間のメッセージ転送時間やストレージの大きさを考慮すべき状況も存在する。これらは将来的には、考慮する必要があるが、本研究の段階では、これらを無視できる状況を仮定している。

DTN 研究の中には、宛先へのメッセージ到着確率を向上させるため、あるメッセージに対し複数のコピーを作成して配送する場合がある。我々の提案手法の中に、この考え方を導入することも考えられるが、現在の研究段階では、メッセージのコピーを作成せずに、メッセージを配送する方式のみを考えている。

ノードの位置およびその時間経過のことを本研究では行動パターンと呼ぶ。実際の環境における行動パターンは極めて複雑であり、本研究の中でそれらを網羅的に扱うことは困難である。そこで本研究の段階では、各ノードがそれぞれ特定の複数地点を巡回する状況を仮定し、その状況下で提案手法を用いると、自律分散的にメッセージの配送が可能になることを示すにとどめる。

本論文では、まず PBR のシステムモデルを定義し、その中でポテンシャルを自律分散的に生成する漸化式を提案、これを TCTR と呼ぶことにする。その後、TCTR の動作および特性を述べる。実験では、プロトタイプシステムとシミュレータを作成し、プロトタイプシステムを使って、(1) ポテンシャルの軌跡やメッセージの配送が期待通り行われるかどうか (すなわち TCTR の実現可能性) を確認し、(2) 実装時に考慮すべきメッセージ氾濫限界 (詳細は第 3.3 節で述べる) の計測を行う。また、シミュレータの妥当性をプロトタイプシステムを使って検証した後、2000[m] × 2000[m] の領域に 100 台のノードを展開し、(3) メッセージ配送時間、(4) ノード間で交換されたメッセージ転送総量、(5) メッセージプールサイズの評価を行う。

以下、本論文の構成を記す。第 2 章で、関連研究を述べる。第 3 章で TCTR システムモデルを提案する。第 4 章で実装方法、第 5 章で評価実験を示す。第 6 章で実験結果などに対する考察を与え、第 7 章でまとめる。

## 2. 関連研究

Delay (or Disruption) Tolerant Networks (DTN)<sup>3)</sup> は、大きな遅延や接続状態が不安定な性質のデータリンク上であっても、メッセージを宛先まで配送することが可能なネットワークである。本研究は、この分野の中でも特にルーティングについて扱っている。ルーティングに対しては、メッセージの配送時間、通信量、バッファサイズ等のコストを小さく抑えることが要求される。

最も、原始的なルーティング手法は、Epidemic Routing<sup>20)</sup> である。メッセージを出会ったノード間で感染させていくことで、最終的に目的地にも届くという楽観的な手法であるため、通信リソースやメッセージプールへの負担は大きい。メッセージの配送時間は最小にすることができる。Spray and Wait 方式<sup>19)</sup> は転送開始時に、一定数のメッセージコピーを作成しておき、それを送り出すことで、到着確率を高くする。一般に、メッセージのコピーを複数作成することで、目的地まで到達する確率を高くし、到着までの時間を短くすることができるが、各種リソースを消費する傾向がある。

ノードの接続関係を、将来にわたって予測し、メッセージの配送経路を最適化しようとする試みがある。ここには、将来の動作が決定的 (i.e., 衛星のように軌道計算が可能) な場合と、非決定的 (i.e., 人間の行動パターンのように予測不可能) な場合によってアプローチが異なる。Merugu らの提案<sup>15)</sup> では、与えられた決定的なノードの行動モデルから時空間グラフを生成し、最小時間配送経路を導き出してメッセージを配送する。一方、将来の動作が非決定的であっても、動作の傾向を学習することでメッセージのルーティング効率を高めるアプローチもある。Leguay らは、MobySpace<sup>10)</sup> と呼ぶ高次元空間に、ノードの行動モデルを対応づけ、ノードの存在確率を計算することで、メッセージを配送する手法を提案した。PROPHET<sup>12)</sup> や Francois<sup>4)</sup> らも、手法は異なるが学習に基づき接続関係を予測する方式を取っている。我々の提案方式は、ポテンシャル値の中に接続関係の情報を取り込むという点で、一種の学習型アルゴリズムと言える。

本研究では、ポテンシャル値を考え、メッセージの配送にこのポテンシャルの勾配を利用する。PBR は勾配探索法 (Gradient Descent Search) の派生系として、Basu らによって、一般のネットワークにおけるトラフィックエンジニアリングの目的で使うことを念頭に提唱された<sup>1)</sup>。その後、移動ネットワークや無線セ

ンサネットワークにおける Anycast での最適ノード選択への利用が試みられている<sup>9)11)</sup>。Volcano Routing Scheme(VRS)<sup>5)</sup>からは、PBR 的な考えがトポロジ変化の激しいネットワークに向いていることを読み取ることができる。PWave<sup>13)</sup>は電子回路における電位と電流の関係をポテンシャルとメッセージの配送に対応づけ、無線センサネットワークに応用した。実フィールドでは、ZebraNet<sup>8)</sup>で提案されている History-based Protocol から、PBR の考え方が具体的な DTN アプリケーションにおいて実際に利用されていることが読み取れる。

### 3. TCTR システムモデル

ここでは提案ルーティング手法である TCTR のシステムモデルを数学的に定義する。本研究での PBR を定義した後、ポテンシャルの生成規則を提案し、この手法の性質を述べる。TCTR を機器に実装する際に考えなければならない氾濫限界と転送スレッシュホルドの関係についても述べる。

本研究では、物理レベルで接続されたネットワークグラフ  $G = (N, E)$  を考える。グラフ  $G$  上のノード  $n \in N$  に対し、 $nbr(n)$  で  $n$  の近隣ノードの集合を表すことにする。いまノード  $n$  において、メッセージの宛先ノード  $d \in N$  について考える。時刻  $t$  における  $d$  宛の  $n$  におけるポテンシャル値を定義し、これを  $V^d(n, t)$  と表記する。 $V^d(n, t)$  はスカラー値であり、それぞれの宛先  $d$  に対して独立な値を持つ。

#### 3.1 Potential-Based Routing(PBR)

時刻  $t$  において  $n$  に存在する宛先  $d$  とラベルづけされたメッセージの集合  $M^d(n, t)$  に対し、近隣ノード  $k \in nbr(n)$  方向に働く力  $F_k^d(n, t)$  を次のように定義する。

$$F_k^d(n, t) \equiv -\{V^d(k, t) - V^d(n, t)\} \quad (1)$$

ここで、 $M^d(n, t)$  の転送先  $nextrhop^d(n, t)$  を、次の規則で与えるのが、本研究の PBR による転送規則である。

$$\forall k \in nbr(n), \\ F_k^d(n, t) \leq \alpha \rightarrow nextrhop^d(n, t) = \Phi \quad (2)$$

$$\exists k \in nbr(n), \\ F_k^d(n, t) > \alpha \rightarrow nextrhop^d(n, t) = \\ \{k | F_k^d(n, t) = \max_{k \in nbr(n)} F_k^d(n, t)\} \quad (3)$$

ここで  $\alpha$  は正の定数で、転送スレッシュホルドと呼ぶことにする。式 2 は、ノードに働く力が  $\alpha$  以下の

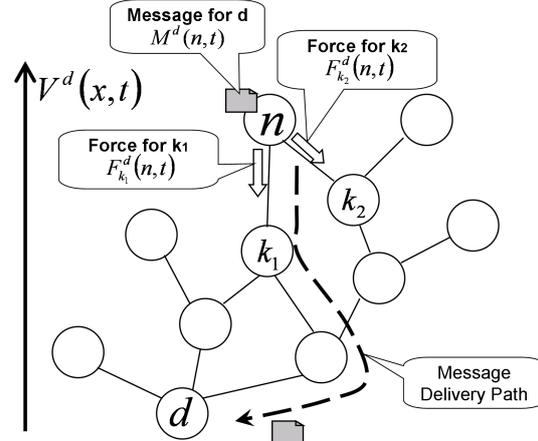


図1 Potential-Based Routing(PBR)  
Fig. 1 Potential-Based Routing(PBR)

時、メッセージの転送先は存在しないことを意味し ( $\Phi$  は空集合)、式 3 は、ノードに働く力が  $\alpha$  を越えれば、最大の力を与える近隣ノードを転送先とすることを意味する<sup>\*1</sup>。この規則で転送を繰り返し行うことで、ポテンシャル値が最も低い宛先  $d$  に、最終的にメッセージが配送される。

図 1 に PBR におけるメッセージ転送および配送の様子を示す。この図では、あるネットワーク  $G$  の物理的なノード間の接続関係を維持しながら、ノードの持つポテンシャル値を縦にとっている。この状態においては、 $n$  の近隣ノード  $k_1, k_2$  で最大の力を与えるものは  $k_1$  であるから、 $n$  にあるメッセージは、 $k_1$  に転送される。同様のことが繰り返し転送先で行われることで徐々に宛先  $d$  へ近づき最終的には届く。メッセージが流れたパスのことを本研究では配送路と呼び、宛先まで届けることを配送と呼ぶ。これに対し、転送は、次のノードへメッセージを送り出す処理である。

PBR のメリットは、近隣ノードとの関係だけでメッセージの転送先を計算できる点である。グローバルなネットワークトポロジに関する情報と必ずしも同期を取らなくてもメッセージの配送が可能である。

#### 3.2 ポテンシャル生成規則

PBR では、ポテンシャルが与えられていれば、式 2,3 に従い、次の転送先を計算することができる。問題はどのようにこのポテンシャルを生成するかであり、本研究では、下記の漸化式により自律的に生成する手法を提案する。

\*1 複数の候補が見つかったときは一つを選び送信するものとする。

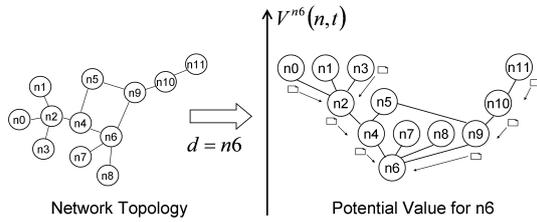


図2 物理的に接続されたネットワークにおける宛先ノード  $d = n6$  へのポテンシャルの形状とメッセージの流れ  
Fig. 2 Potential formation and message delivery path for destination  $d = n6$  in a physically connected network.

$$V^d(n, t+1) = V^d(n, t) + D \min_{k \in nbr(n)} \{0, V^d(k, t) - V^d(n, t)\} + \rho \quad (4)$$

$$\forall d \in N \forall t V^d(d, t) = 0 \quad (5)$$

$$\forall n \in N V^d(n, 0) = 0 \quad (6)$$

これは  $V^d(n, t)$  に対して、自分の持つポテンシャルよりも低い最小のポテンシャルを持つノードが近隣にあれば、「そのノードとのポテンシャル差分に定数  $D (0 < D < 1)$  をかけたもの」の分だけ自分のポテンシャルを下げ、「定数  $\rho (0 < \rho < D)$ 」分だけ自分のポテンシャルを上げるようにするということである。式5, 式6は拘束条件で、それぞれ、宛先が自分自身であるならばポテンシャル値は0, 開始時のポテンシャル値は0, を表している。

このような形式でポテンシャルを定義すれば、 $V^d(n, t)$  の性質を力学的な分析から明らかにすることができる。以下では、3種類のネットワーク状態においてポテンシャルがどのような形を取るかを述べる。

- 物理的に接続された単一ネットワークの場合 (図2)  
ポテンシャル値は収束し、距離ベクトル型のルーティングと等価な状態になる。図2左側のネットワークに対して生成される、宛先  $d = n6$  へのポテンシャル  $V^{n6}(n, t)$  の収束値を図2の右側に示す。n6からホップ数が増えるごとにポテンシャル値は増加し、メッセージはポテンシャルの低い方向へ配送される。
- ネットワークが物理的に分裂した場合 (図3)  
ネットワークが二つに分裂すると、宛先ノードが属さないネットワーク・ノードのポテンシャル値が増加を開始する。一般に、分裂したネットワークを  $G_1 = (N_1, E_1), G_2 = (N_2, E_2)$  とし、 $G_1$  に宛先ノードが存在する場合 (i.e.,  $d \in N_1$  であるとき),  $N_2$  のノードのポテンシャルは下記に収束

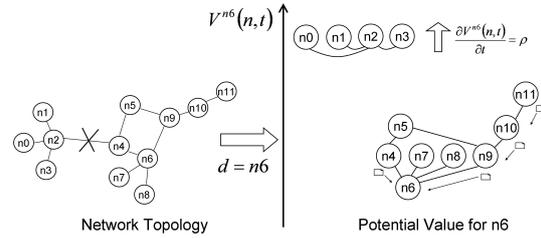


図3  $n2 - n4$  間のリンクが切断され、二つのネットワークに分離してしまった場合のポテンシャル形状;  $n0, n1, n2, n3$  のポテンシャルは上昇を開始し、その速度は  $\rho$  に収束する。

Fig. 3 Potential formation when  $n2 - n4$  link has been disrupted. Potentials of  $n0, n1, n2, n3$  goes up with the velocity of  $\rho$ .

する ( $c$  は定数)。

$$\forall n \in N_2 V^d(n, t) \rightarrow \rho t + c \quad (7)$$

$$(t \rightarrow \infty)$$

図3の例では、 $\{n0, \dots, n3\}$  のポテンシャルは  $\rho$  の速度で上昇する。このとき、 $\{n0, \dots, n3\}$  にあるメッセージは、各ノードのメッセージプールに蓄えられる。

- 物理的に分裂した二つのネットワークが再結合する場合 (図4)  
図4では、 $n1 - n5$  間のリンクが接続されたことにより、二つのネットワークが再結合している。ポテンシャルの低い方のネットワークに接続したノードから順にポテンシャルが降下し、図4のようなポテンシャルのカーブが生ずる。ここを伝わって  $\{n0, \dots, n3\}$  のメッセージプールに蓄えられていたメッセージが一気に流れ出る。

ポテンシャルは宛先への到達性を表す、ある種の指標である。「ポテンシャルがより小さい」と「宛先ノードにより効率よくメッセージを配送できる」が等価となるネットワーク環境であれば、本提案手法は最大のパフォーマンスを発揮する。

実際、下記の状況が仮定できる環境 (i.e., ノードの接続関係, 行動パターン) であれば、式4から生成されるポテンシャル形状は、メッセージ配送に関して最適な形となる。

- 宛先  $d$  の存在するネットワークから離れているよりは、接続されている方がメッセージは早く配送される環境
- より近い過去に  $d$  の存在するネットワークから離れたものが、より近い未来に再接続される確率が高い環境

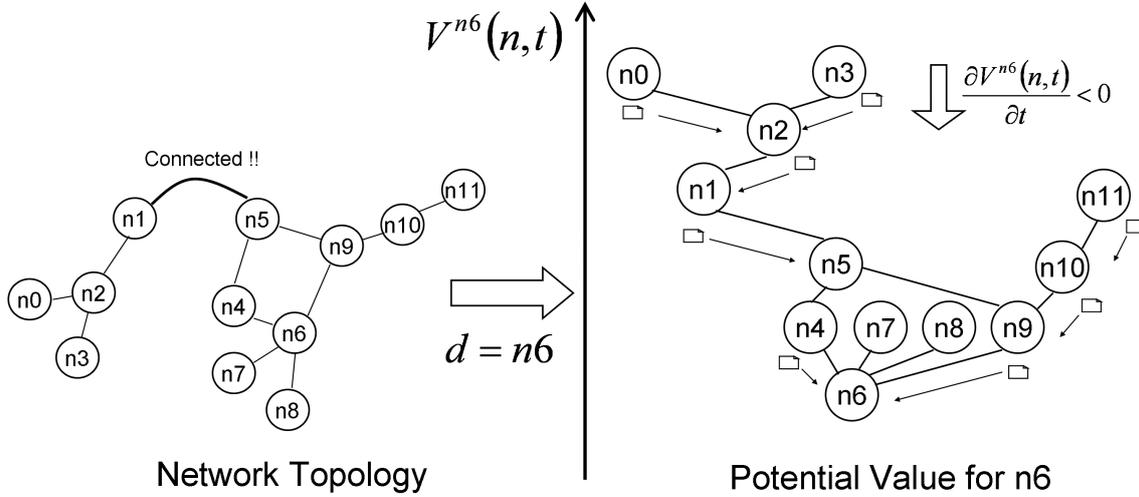


図 4  $n1 - n5$  間のリンクが接続されたことにより、二つのネットワークが再結合する場合.  $n0, n1, n2, n3$  のポテンシャルが急降下すると共に、カーブが生まれ、メッセージが流れていく.

Fig. 4 When  $n1 - n5$  link has been setup, potentials of  $n0, n1, n2, n3$  goes down with making message delivery curves.

### 3.3 ポテンシャル伝播遅延とメッセージ氾濫現象

式 4 では、近隣ノード  $k \in nbr(n)$  が持つポテンシャル  $V^d(k, t)$  も時刻  $t$  で取得できていることになっているが、実際に TCTR を実装する場合には、ノード間で定期的にポテンシャルを交換するため、ノード  $n$  に見える  $k$  が持つポテンシャル (以下、 $V_n^d(k, t)$  と記述する) は、現時点での  $k$  が持つポテンシャル  $V_k^d(k, t)$  よりも少し古いものになる. この時間差を  $\Delta T$  とすると、変化したポテンシャル量に関して、

$$\begin{aligned} V_k^d(k, t) - V_k^d(k, t - \Delta T) \\ = V_k^d(k, t) - V_n^d(k, t) \end{aligned} \quad (8)$$

が成立する. 従って、

$$V_n^d(k, t) \approx V_k^d(k, t) - \Delta T \frac{\partial V_n^d(n, t)}{\partial t} \quad (9)$$

となり、ポテンシャル値の時間変化によって、観測される近隣ノードのポテンシャル値に変位が生じる. この変位が式 4 だけからは考えられなかった、メッセージの氾濫と呼ぶ現象を引き起こすことがある.

簡単な状況として、 $\forall i \in N, V_i^d(i, t) = V(t)$  という、すべてのノードが平衡に同一ポテンシャル値  $V(t)$  を持つ場合を想定しよう. これは、図 3 に示したように分離されたネットワークで実際に存在しうる状況である. 式 2,3 において  $\alpha = 0$  の場合、理想的な環境下では  $nexthop^d(n, t) = \Phi$  となり、転送先は存在しない. しかし、実際的な環境では、式 7 が成り立つため、

$$V_k^d(n, t) \approx V(t) - \Delta T \frac{\partial V(t)}{\partial t} \quad (10)$$

すなわち、式 1 より、

$$F_k^d(n, t) = \Delta T \frac{\partial V(t)}{\partial t} \quad (11)$$

となつて、ポテンシャル値が増加傾向にあるとき ( $\Leftrightarrow \frac{\partial V(t)}{\partial t} > 0$ ) は、式 3 から、

$$nexthop^d(n, t) = \bigcup_{k \in nbr(n)} k \quad (12)$$

となり、すべてのノードが近隣ノードの 1 つを選んでメッセージを転送する状態になる. 結果、メッセージがネットワークの中を巡り続ける状況となってしまう. 本研究では、この現象をメッセージの氾濫現象と呼んでいる.

具体的な場面として、図 5 の状況を取り上げる. 開始時には、ノード  $A, B, C$  はすべて相互に接続されていて、 $A, B$  における宛先  $C$  のメッセージ転送先は、それぞれ  $A \rightarrow C$  と  $B \rightarrow C$  である. ある時点で、 $C$  が切り離され遠くに行ってしまったとする. このとき、 $A, B$  の  $C$  へのポテンシャル値は、共に同じ値で増加を開始する.  $A$  が見る  $B$  のポテンシャル値は、 $B$  の本来のポテンシャル値より小さい. そこで  $A$  は、 $C$  へのメッセージを  $B$  へ送り出そうとする. 同様のことは、 $B$  にも当てはまり、 $B$  は  $C$  へのメッセージを  $A$  へ送り出そうとする. そのため、メッセージはネットワーク内を巡り続けるのである.

この問題現象を回避するために、実際のネットワー

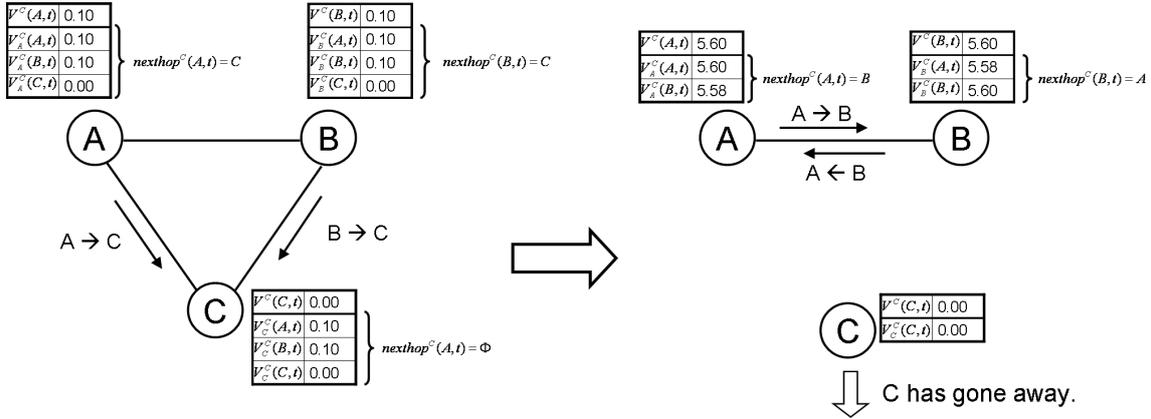


図 5 メッセージ氾濫現象の一例

Fig. 5 An example of message overflow phenomenon

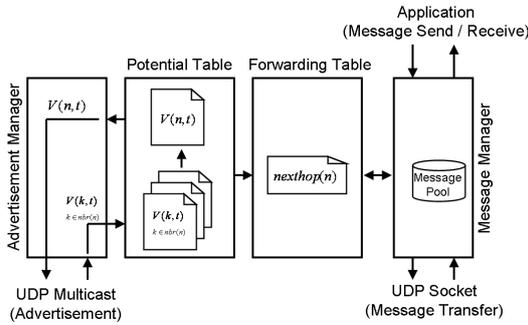


図 6 プロトタイプシステムの構成

Fig. 6 Prototype System Overview

クでは、式 2,3 で定義した転送スレッショールド  $\alpha$  を設定する．設定においては、氾濫限界  $\beta$  を考え、

$$\beta < \alpha \quad (13)$$

となるように  $\alpha$  を設定する．氾濫限界  $\beta$  は式 9 より、

$$\beta = \Delta T \max \frac{\partial V^d(n, t)}{\partial t} \quad (14)$$

で与えられ、式 4 によれば、

$$\max \frac{\partial V^d(n, t)}{\partial t} = \rho \quad (15)$$

だから、

$$\beta = \rho \Delta T \quad (16)$$

である．

#### 4. 実 装

本研究では、実際に無線端末で動作するプロトタイプシステムと、仮想的に大規模なシナリオを再現できるシミュレータを開発した．

##### 4.1 プロトタイプシステム

図 6 に本研究で開発したプロトタイプシステムの構

成を示す．4 種の機能ブロックで構成してある．以下、このシステムがノード  $n$  として動いている状況を考え、各機能を解説する．

- (1) Advertisement Manager: UDP Multicast により、定期的に近隣ノード  $k \in nbr(n)$  とポテンシャル  $V(k, t), V(n, t)$  を交換しあう．Multicast Socket を通じ、近隣ノードから受け取ったポテンシャル  $V(k, t)$  は、PotentialTable に送られる．一方、PotentialTable からは定期的に、このノード  $n$  の最新ポテンシャル  $V(n, t)$  が与えられ、これを UDP Multicast で広告する．
- (2) Potential Table: すべての近隣ノードのポテンシャル  $\forall k \in nbr(n) V(k, t)$  をキャッシュし、このノードのポテンシャル  $V(n, t)$  とあわせて  $V(n, t+1)$  を式 4 に従い計算する．
- (3) Forwarding Table: Potential Table から  $V^d(n, t)$  と  $\forall k \in nbr(n) V^d(k, t)$  を与えられ、式 2, 3 に従って、宛先  $d$  ごとに転送先ノードを計算し、メッセージ転送表を作成する．
- (4) Message Manager: アプリケーションへのメッセージインタフェースを提供し、メッセージの転送処理も行う．アプリケーションは、Message Manager に対して、メッセージの配送を委託する (Send)．Message Manager が  $n$  宛のメッセージを受信すれば、アプリケーションへ通知する (Receive)．通常は、 $n$  が受信したメッセージに対して、宛先  $d$  をキーに Forwarding Table に転送先を問合せ、転送先が存在すればメッセージを転送する．転送が成功すると転送先から応答として ACK が通知される．転送先が存在しない場合や、転送に失敗したときは、その

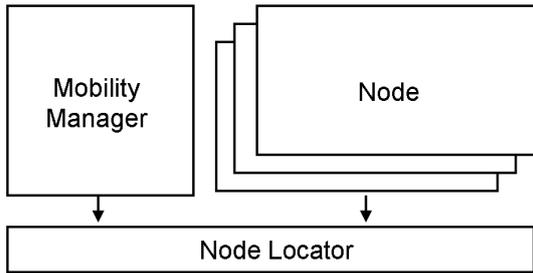


図 7 シミュレータの構成  
Fig. 7 TCTR Simulator Overview

メッセージは Message Manager のメッセージプールに保存される．保存されているメッセージは，定期的に再転送が試みられる．

このプロトタイプシステムは，下記内容をログとして出力する．このログは，次の評価実験での解析に利用した．

- ポテンシャル値 (定期的にログ出力)
- Forwarding Table の内容 (定期的にログ出力)
- メッセージプールサイズ (定期的にログ出力)
- メッセージ転送イベント
- メッセージ送信 (Send) イベント
- メッセージ受信 (Receive) イベント

ここで開発したプロトタイプシステムは C 言語で実装した．実験環境は，OS が Linux 2.6.22 で，無線データリンクには 802.11g を使用している．

#### 4.2 TCTR シミュレータ

図 7 に，開発したシミュレータの構成を示す．3 種の機能ブロックで構成してある．以下，それぞれの機能ブロックについて解説する．

- (1) Node: 各 Node インスタンスが，仮想的に展開されたノードに相当し，独立したスレッドにより動作する．Node の内部は，プロトタイプシステムとほぼ同様の構成となっている．ただし，Node Locator を通じて，近隣ノードへのポテンシャル通知 (広告) やメッセージ交換を行う．
- (2) Node Locator: Node インスタンスの (2 次元平面での) 位置管理を行う．通信可能範囲にあるノードの発見機能，位置の操作インタフェースなどを持つ．
- (3) Mobility Manager: 実験対象とする行動パターンを生成し，Node Locator を通じてノードの位置設定をする．

出力ログは，プロトタイプシステムと同様に下記を出力する．このログを評価実験での解析に使用した．

- ポテンシャル値 (定期的にログ出力)
- Forwarding Table の内容 (定期的にログ出力)

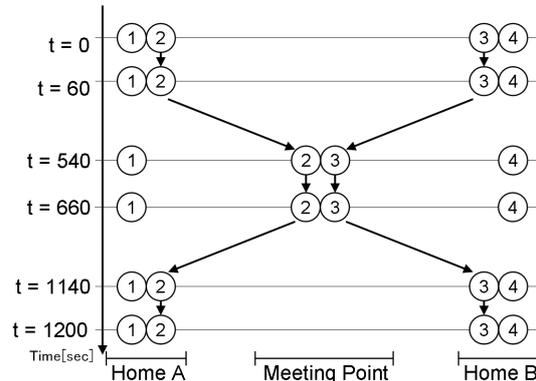


図 8 実証実験におけるノードの行動パターン  
Fig. 8 Mobility model used in the experiment

- メッセージプールサイズ (定期的にログ出力)
- メッセージ転送イベント
- メッセージ送信 (Send) イベント
- メッセージ受信 (Receive) イベント

本研究で開発したシミュレーション環境は，Java 言語で実装した．実行には，JavaVM 1.5.0，OS は Linux 2.6.15 を使用した．

## 5. 評価

次に述べる評価実験では，まず，プロトタイプシステムを使った実証実験により本提案システムが実現可能であることを確認し，シミュレータが実証実験での結果を再現できることを確認した．次にプロトタイプシステムによりメッセージ氾濫現象を観測，分析した．シミュレータによる大規模なシナリオの検証では，配送時間が最小となる Flooding-Based Routing(FBR)方式と次の評価基準について比較を行った: (1) 配送時間，(2) メッセージ転送の総量，(3) メッセージプールサイズ (プールに保存されているメッセージ数)．

下記実験は， $D$  および  $\rho$  のパラメータに，それぞれ  $D = 0.2$ ， $\rho = 0.02$  を設定して行われた．

### 5.1 プロトタイプシステムによる実証実験およびシミュレータの妥当性

この実験では，物理的に隔離されたネットワーク間をノードが行き来するシナリオにおいて，プロトタイプシステムによる動作の確認を行った．具体的には，ポテンシャル変化の様子やメッセージ配送の様子を観測し，それが期待通りのものであるか調べた．その後，シミュレータで同様のシナリオを設定し，ほぼ同じ結果を得られることを確認した．

図 8 にこの実験で利用したノードの行動パターンを示す．Home A，Home B，Meeting Point を考え，

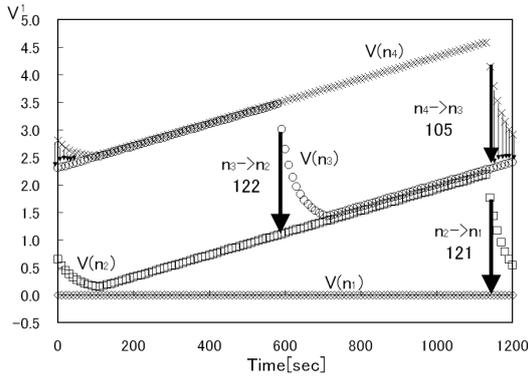


図9 プロトタイプシステムによるポテンシャル変化とメッセージ配送の様子

Fig. 9 Potential patterns with the prototype-based experiment

ノード 1, 4 はそれぞれ Home A, B の固定ノード, ノード 2, 3 は, Home A, B の住人が持つ移動ノードとする. ノード 2, 3 は, Meeting Point で定期的に出会うことになっている. Home A, Meeting Point, Home B は十分に離れており, 直接通信することはできない. 無線の通信範囲は (建造物等による影響を考慮) 20m ほどを想定している. 図 8 に示すように, ノード 2, 3 を行動させた. 各所への滞在時間は 2 分である. 移動中は無線通信範囲を一度出ると, どのノードとも通信はできない. 図 8 に示した移動のプロセスは 20 分を周期に, 繰り返される. この実験においては, ノード 4 がノード 1 宛に, 10 秒に 1 回, 100 バイト程度のメッセージを送信している.

図 9 に, プロトタイプシステムで実験した結果得られた, 各ノードにおける (ノード 1 への) ポテンシャル値  $V^1(n, t)$  の挙動と, メッセージ配送の様子を示す. 実験開始後, 時刻  $t = 3600$  から  $t = 4800$  までの様子を示しているが, 図の時間軸は, わかりやすくするため ( $t - 3600$ ) を表示している. ここで,  $n_1, n_2, n_3, n_4$  はそれぞれ, ノード 1, ノード 2, ノード 3, ノード 4 に対応している.

時刻  $t = 60$  を過ぎて, しばらくすると  $n_2$  は  $n_1$  が近隣ノードでなくなったことを知り,  $V(n_2, t)$  は上昇を開始した.  $t = 580$  において,  $n_3$  は  $n_2$  と接続関係になったことを知り,  $V(n_3, t)$  を降下させると共に, 保有しているメッセージ 122 通を  $n_2$  に転送した. その後,  $n_2$  と  $n_3$  は Meeting Point を離れ,  $t = 1140$  において,  $n_4$  は  $n_3$  と,  $n_2$  は  $n_1$  と接続関係になったことを知った. 結果,  $n_4, n_2$  はポテンシャルを降下させながら, それぞれ保有している  $n_1$  宛のメッセージ 105 通と 121 通を  $n_3, n_1$  に転送した.

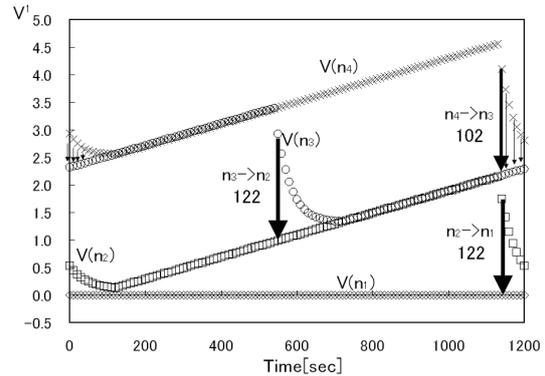


図10 シミュレータによるポテンシャル変化とメッセージ配送の様子

Fig. 10 Potential patterns with the simulation-based experiment

このメッセージ配送は自発的に行われた. そして, この動作は期待通りのものであった. こうして TCTR ルーティングが実現可能であることが実証された.

図 10 に, シミュレータで実験した結果得られた, 各ノードにおける (ノード 1 への) ポテンシャル値  $V^1(n, t)$  の挙動と, メッセージ配送の様子を示す. プロトタイプシステムの場合と同様に, 時刻  $t = 3600$  から  $t = 4800$  までの様子を示した (時間軸は ( $t - 3600$ ) を表示している). 結果は, プロトタイプシステムの場合と, ほぼ同じだった.

こうして, シミュレータがプロトタイプシステムでの実験を再現できることを確認した.

## 5.2 メッセージ氾濫限界の計測

ここでは図 5 のシナリオで, 様々な転送スレッシュホルドを設定し,  $C$  宛のメッセージの転送先がどのように収束するかを観測した.

式 16 によると, 実験の環境  $0 < \Delta T < 1, \rho = 0.02$  においては, 氾濫限界は, 0.02 未満であることが予測される. 実際に  $\alpha$  を  $\{0.000, 0.010, 0.015, 0.020, 0.025, 0.030\}$  に設定し, 図 5 の状態を作ることで, 生成される転送先を調べた.

表 1 に実験結果を示す. 各  $\alpha$  に対して 3 回ずつ図 5 のシナリオを再現することで, Node A での転送先と Node B での転送先を調べた. この表には, その結果および, その結果からわかるネットワークの状態 (Status) が記載されている. "A" は転送先が Node A, "B" は転送先が Node B, "Φ" は転送先が存在しない, ことを意味する. 状態について, "ループ" は, メッセージがネットワーク内を巡回する状態になったことを意味し, "偏り" はいずれか片方のノードにメッセージが集まる状態になったこと, "ホールド" はメッ

表 1 メッセージ氾濫限界の計測結果  
Table 1 An Analysis of the Message Overflow Threshold

$\alpha$	Node A での方向			Node B での方向			Status
	1 回目	2 回目	3 回目	1 回目	2 回目	3 回目	
0.000	B	B	B	A	A	A	ループ
0.010	$\Phi$	B	B	A	A	$\Phi$	偏り ループ
0.015	$\Phi$	$\Phi$	B	A	A	$\Phi$	偏り
0.020	$\Phi$	B	B	$\Phi$	$\Phi$	$\Phi$	偏り ホールド
0.025	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	ホールド
0.030	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	ホールド

セージは転送されない状態になったことを意味している。複数の状態が観測された  $\alpha$  に対しては、すべての状態を記載した。

”偏り”の状態は、片方のノードが氾濫現象を起こしていることを表している。 $\alpha$  が 0.02 では氾濫現象はほぼ見られなくなり、この値を越えると現れなくなっている。

### 5.3 配送時間, 転送総量, プールサイズ

この実験では、シミュレータを用いて、100 台規模のノードが存在する環境を仮想的に作り出し、メッセージ配送時間、メッセージ転送総量、メッセージプールサイズに関して評価を行っている。

評価においては、Flooding-Based Routing(FBR)との比較を行った。実験に用いた FBR は、次の方法でメッセージの配送を行う。いま、ノード  $A(\in N)$  とノード  $B(\in N)$  を考え、 $A, B$  が保持しているメッセージの集合をそれぞれ  $M_A, M_B$  とする。ノード  $A$  と  $B$  が出会ったときに、次の規則でメッセージ交換を行う。

$$m \in M_A \wedge \neg m \in M_B \rightarrow Transfer_A(m, B) \quad (17)$$

$$m \in M_B \wedge \neg m \in M_A \rightarrow Transfer_B(m, A) \quad (18)$$

ここで、 $Transfer_X(m, Y)$  は、ノード  $X$  がメッセージ  $m$  をノード  $Y$  に転送することを意味する。結果、

$$\forall m(m \in M_A \leftrightarrow m \in M_B) \quad (19)$$

となる。各メッセージ  $m$  には、宛先ラベルがついていて、宛先に到達したらそのメッセージはアプリケーションに届けられる。FBR 方式は、メッセージの配送にかかる時間を最小化することができる。本研究では、FBR の配送時間と TCTR の配送時間を比較することで、どの程度の効率があるかを調査した。いずれの場合もメッセージの Time To Live(TTL) は無限大に設定している。

実験では、次のようにノードの行動パターンを設定した (FBR と TCTR で同じ行動パターンを利用した)。

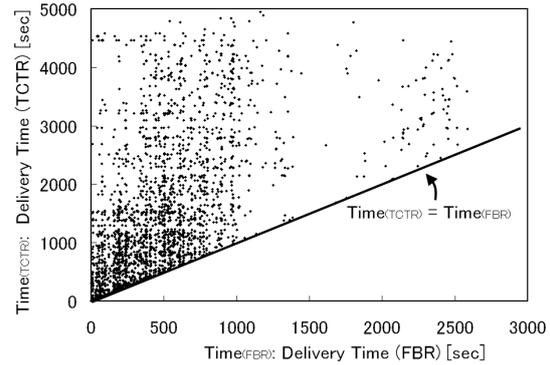


図 11 転送時間の関係

Fig. 11 Delivery Time Relationship between FBR and TCTR

2000[m] × 2000[m] の 2 次元平面に、半径 0 ~ 300[m] の円運動をするノードを 100 個配置した。無線通信可能半径は 150[m] で、回転周波数は、0 ~ 0.01[Hz] である。行動パターンとしては、Random Way Point(RWP)<sup>2)</sup> も考えられるのだが、本研究では、ノードが巡回的に行動する状況を考えているため RWP での評価は行っていない。

時刻  $t=0$  から動作を開始し、 $t=10000$  にすべてのノードがすべてのノードに対して一斉にメッセージを送信する ( $t=10000$  は、TCTR においてポテンシャル情報が全体に行き渡るために十分な時間である)。

#### 5.3.1 メッセージ配送時間

図 11 に、FBR 方式での配送時間  $Time_{(FBR)}$  と TCTR 方式での配送時間  $Time_{(TCTR)}$  の関係を示す。それぞれの点は (送信元:宛先) のペアに対応する。この図には、 $Time_{(TCTR)} = Time_{(FBR)}$  である線を書き込んである。各点は、 $Time_{(TCTR)} \geq Time_{(FBR)}$  の領域に存在することが読み取れる。

図 12 に、配送時間の倍率  $\frac{Time_{(TCTR)}}{Time_{(FBR)}}$  と、その倍率に属したメッセージ配送処理の分布を示す。この図から、約 33% のメッセージが最適 (FBR) の場合の 2 倍時間以内に配達されており、約 50% のメッセージが最適の場合の 3 倍時間以内に配達されていることが読

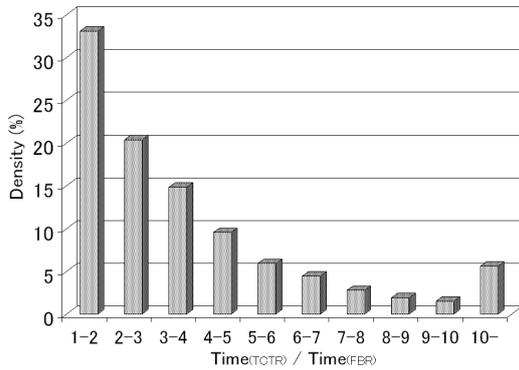


図 12 配送時間の倍率とその分布

Fig. 12 Efficiency in Delivery Time and its Distribution

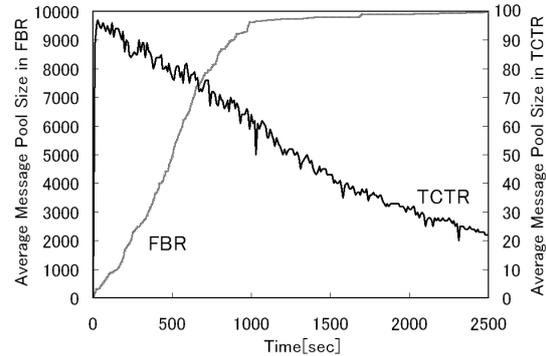


図 14 メッセージプールの平均サイズ

Fig. 14 The Average Message Pool Size

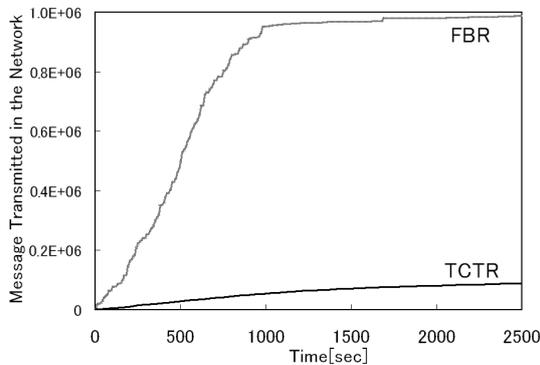


図 13 メッセージ転送総量の変化

Fig. 13 The Total Message Transmission in the Network

み取れる．10 倍時間以上かかる配送処理は，全体の 5%程度であった．

### 5.3.2 メッセージ転送総量

図 13 に，FBR 方式と TCTR 方式でのメッセージ転送総量を示す（時刻は，メッセージ配送を開始した  $t = 10000$  を 0 として表示している）．転送総量とは，任意のノード  $a \in N$  からノード  $b \in N, a \neq b$  へメッセージ転送があった場合にカウントされるグローバルな変数である．この図から読み取れるように，FBR 方式では，TCTR に比べておよそ 10 倍のメッセージ転送を行った．FBR は，すべてのノード（=100 台）それぞれが 100 個のノードから発される 100 箇所へのメッセージ転送処理すべてに加担するため，最終的に全体で約  $10^6$  回の転送を行ったと読み取れる．一方，TCTR では，各ノードは，100 個のノードから発される 100 箇所へのメッセージ転送の一部のみを行ったため，約  $10^5$  回の転送で済んだものと思われる．

### 5.3.3 メッセージプールの大きさ

図 14 に，FBR と TCTR における平均メッセージプール量の時間変化を示す（時刻は，メッセージ配送を開始した  $t = 10000$  を 0 として表示している）．FBR では，メッセージプールの大きさが時間の経過と共に増加し 10000 で安定しようとしている．一方，TCTR では，メッセージプールが最初 97 まで増加したが，時間の経過と共に減少している．

FBR では，すべてのノードがネットワークに存在するすべてのメッセージをメッセージプールに保存する．100 個のノードから，100 箇所へメッセージが配送されたので， $100 \times 100 = 10000$  に収束したのだと考えられる．TCTR の場合はメッセージは複製されず配送が完了すればネットワークから消えていくため減少したのだと考えられる．

## 6. 考 察

本研究では，プロトタイプシステムによる実機での動作検証と，シミュレータによる大規模な状況を想定した評価実験を行った．同じ行動パターンを用意すれば，プロトタイプシステムでのポテンシャルやメッセージ配送の挙動を，シミュレータによる実験で再現できた．これはシミュレーション実験の結果にある程度の妥当性を与えるものである．

プロトタイプシステムを使って実際にメッセージ氾濫現象を観測し，氾濫限界  $\beta$  が 3.3 節で与えたモデルに従っていることが確認された．これにより TCTR を実環境に展開するときに，転送スレッシュホールド  $\alpha$  の適性値を設計することが可能になったと言える．

シミュレータによる 100 台規模の環境の評価では，配送時間が最短となる FBR との比較を行った．結果，実験の環境では，およそ半数のメッセージは FBR の

3倍時間以内に配送されることが観測された。ここで TCTR に対して、FBR は通信量にしておよそ 10 倍、メッセージバッファ量に関しては 100 倍以上のコストがかかっている。配送時間の遅れが十分許容できるのであれば、TCTR を使ったほうが、リソースの消費が少なく済むと言える。

本研究では、巡回的な行動パターンについてのみ扱い、一般的な行動パターンについては扱わなかった。また、メッセージ単位の大きさ、通信帯域、トラフィックの輻輳を考慮したメッセージ転送などは無視していた。メッセージ単位が大きく、通信帯域の狭い通信であれば、一つのメッセージの転送にかかる時間が無視できなくなってしまう<sup>7)</sup>。将来的には、このような状況も想定する必要があると考えている。

## 7. おわりに

本研究では、DTN 環境向けに、トポロジ変化に強いルーティング手法である TCTR を提案した。メッセージの転送に PBR の勾配法を用い、ポテンシャルの生成を自律分散的に行う、時間に関する漸化式を定義した。

プロトタイプシステムを開発し、TCTR が実際に実装可能であることを示した。同時に、実装時に想定しなければならないメッセージ氾濫現象について言及およびモデル化し、評価試験では、このモデルの検証も行った。

TCTR を用いると、特に巡回的にノードが行動するシナリオにおいて、自律的にポテンシャルを生成し、メッセージを効率よく配送できた。評価実験で述べた 100 台のノードが行動する環境において、配送時間が最小となる FBR 方式と比べ、約半数のメッセージが 3 倍以内の時間で配送され、メッセージ転送総量は 10 分の 1、メッセージプールサイズは 100 分の 1 以下にまで抑えられた。

## 参考文献

- 1) Basu, A., Lin, A. and Ramanathan, S.: Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm, *ACM SIGCOMM 2003*, pp.37–48 (2003).
- 2) Bettstetter, C., Resta, G. and Santi, P.: The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks, *IEEE Transactions on Mobile Computing*, Vol.2, No.3, pp.257–269 (2003).
- 3) Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets, *ACM SIGCOMM 2003*, pp.27–34 (2003).
- 4) Francois, J.-M. and Leduc, G.: Delivery Guarantees in Predictable Disruption Tolerant Networks, *Lecture Notes in Computer Science*, Vol.4479, pp.167–178 (2007).
- 5) Ganjali, Y. and McKeown, N.: Routing in a Highly Dynamic Topology, *IEEE SECON*, pp.164–175 (2005).
- 6) Iren, S., Amer, P.D. and Conrad, P.T.: The Transport Layer: Tutorial and Survey, *ACM Computing Surveys*, Vol.31, No.4, pp.360–404 (1999).
- 7) Jain, S., Fall, K. and Patra, R.: Routing in a Delay Tolerant Network, *ACM SIGCOMM 2004*, pp.145–158 (2004).
- 8) Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L.-S. and Rubenstein, D.: Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet, *ACM SIGOPS*, pp.96–107 (2002).
- 9) Kumar, P., Kuri, J., Nuggehalli, P., Strasser, M., May, M. and Plattner, B.: Connectivity-aware Routing in Sensor Networks, *IEEE SensorComm*, pp.14–20 (2007).
- 10) Leguay, J., Friedman, T. and Conan, V.: DTN Routing in a Mobility Pattern Space, *ACM SIGCOMM workshop on Delay-tolerant networking*, pp.276–283 (2005).
- 11) Lenders, V., May, M. and Plattner, B.: Density-based vs. Proximity-based Anycast Routing for Mobile Networks, *IEEE INFOCOM*, pp.1–13 (2006).
- 12) Lindgren, A., Doria, A. and Schelen, O.: Probabilistic Routing in Intermittently Connected Networks, *Lecture Notes in Computer Science*, Vol.3126, pp.239–254 (2004).
- 13) Liu, H., Zhang, Z.-L., Srivastava, J. and Firoiu, V.: PWave: A Multi-source Multi-sink Anycast Routing Framework for Wireless Sensor Networks, *Lecture Notes in Computer Science*, Vol.4479, pp.179–190 (2007).
- 14) Malkin, G.: RFC2453: RIP Version 2 (1998).
- 15) Merugu, S., Ammar, M. and Zegura, E.: Routing in Space and Time in Networks with Predictable Mobility, Technical report, Georgia Institute of Technology (2004).
- 16) Moy, J.: RFC2328: OSPF Version 2 (1998).
- 17) Perkins, C., Belding-Royer, E. and Das, S.: RFC1058: Ad hoc On-Demand Distance Vector (AODV) Routing (2003).
- 18) Rosen, E., Viswanathan, A. and Callon, R.: RFC3031: Multiprotocol Label Switching Architecture (2001).
- 19) Spyropoulos, T., Psounis, K. and Raghaven-

- dra, C.S.: Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks, *ACM SIGCOMM workshop on Delay-tolerant networking*, pp.252–259 (2005).
- 20) Vahdat, A. and Becker, D.: Epidemic Routing for Partially-Connected Ad Hoc Networks, Technical report, Duke University (2000).
-