

MANET における物理ネットワークを考慮した オーバーレイネットワーク構築手法

西原 雄太[†] 陶山 優一^{††} 植田 裕規^{††} 横田 裕介[†] 大久保 英嗣[†]

[†]立命館大学情報理工学部 ^{††}立命館大学大学院理工学研究科

1 はじめに

近年、PC のみならず、携帯電話、PDA、ビデオカメラ、センサなどのあらゆる機器に無線デバイスが搭載され、互いに通信することで人の社会生活をサポートするユビキタス環境が実現されつつある。その中で、アクセスポイントやルータを必要とせず、無線で接続できる端末によって、自律的に構成されるモバイルアドホックネットワーク(以下、MANET と記す)がインフラストラクチャとして注目されている。しかし、MANET では、ブロードキャストを使用してリソースや経路の発見を行うため、大規模なネットワークを構築できない。また、一般に携帯機器は電力資源に乏しく、省電力性を考慮しなければならない。

これらの問題を解決するために、特定のサーバが存在しない環境において、高速な検索を可能とする技術である分散ハッシュテーブル(以下、DHT と記す)を利用することが考えられている。DHT は、広範なインターネットアプリケーションにおいて、オブジェクトの高速発見や負荷分散に利用されており、MANET に適用させた場合でも同様に効率性が期待される。しかし、DHT では物理ネットワークにおけるノード間の近接性を考慮せずオーバーレイネットワークを構築するという問題がある。

本稿では、物理ネットワークのノード間の近接性を考慮したオーバーレイネットワークを構築手法を提案する。これにより、ネットワークの負荷の軽減や電力消費の軽減が可能となる。

2 関連研究

MANET 環境においてノード間の近接性を考慮したオーバーレイネットワークの構築を行っている研究に MADPastry[1]がある。MADPastry ではネットワークをクラスタリングによって分割し、クラスタ内のノードに近いノード ID の割り振りを行う。これにより、物理ネットワークにおけるホップ数を削減し、トラフィックを軽減させている。

しかし、MADPastry ではクラスタリングを行う際にクラスタヘッドを必要とするため、クラスタヘッドへの負荷集中が起こる。また、ネットワークが大規模化してクラスタ内のノードが増えた場合、クラスタヘッドへの負荷の増大やネットワーク全体のパフォーマンスが低下するといった問題がある。このため、ネットワークの規模を事前に把握し、規模に応じてクラスタヘッドの数を設定しなければならない。

そこで、本手法ではクラスタリングを用いず、近隣のノードの情報から自身のノード ID を決定する。これにより、クラスタヘッドへの負荷集中やネットワークの規

模に応じた設定の必要がなくなる。

3 提案手法

本手法では、近隣のノードからノード ID や DHT の経路情報を取得し、その情報を基にして自身のノード ID を決定する。本手法では、代表的な DHT アルゴリズムの一つである Chord[2] を拡張する。Chord は環状のハッシュ空間上で二分探索に似た探索を行うことで $\log(N)$ の探索効率を実現したルーティングアルゴリズムであり、確実な探索とノードの参加・離脱に対応している。

本手法は、ノード ID の決定と再計算、経路情報の維持管理、レプリケーションの 3 つに大きく分類される。以下、本章ではそれぞれの機構について述べる。

3.1 ノード ID の決定と再計算

近隣ノードからの情報を基にし、Chord の環状ハッシュ空間における自身の前のノード (Predecessor ノード) と次のノード (Successor ノード) とのホップ数が短くなるように自身の ID を決定する。本手法によってノード ID が割り振られるまでの処理の流れを次に示す。

近隣ノードから情報を取得

自ノードと 1 ホップで繋がっている近隣ノードに対してクエリを送信し情報を取得する。取得する情報は、ノード ID、Successor・Predecessor ノード ID、物理ネットワークにおける Successor・Predecessor ノードまでの経路、ノードの移動性である。ノードの移動性は、ノード ID が決定してからノード ID の再計算が起こるまでの経過時間から求める。

ノード ID 候補を計算

取得した情報を基にしてノード ID 候補を得る。まず、任意の近隣ノードの Successor・Predecessor ノードか Successor・Predecessor ノードまでの経路にあるノードが、近隣ノードの中に存在するかを調べる。Successor ノードか Successor ノードへの経路にあるノードが存在していた場合は、以下の式を使ってノード ID 候補を得る。

$$\left(\text{近隣ノード ID} + \frac{\text{Successor ノード ID} - \text{近隣ノード ID}}{2} \right) \bmod \text{ハッシュ値の最大値} = \text{ノード ID 候補} \quad (1)$$

Predecessor ノードか Predecessor ノードへの経路にあるノードが存在していた場合は同様に、以下の式を使ってノード ID 候補を得る。

$$\left(\text{Predecessor ノード ID} + \frac{\text{近隣ノード ID} - \text{Predecessor ノード ID}}{2} \right) \bmod \text{ハッシュ値の最大値} = \text{ノード ID 候補} \quad (2)$$

これをすべての近隣ノードに対して行い、ノード ID 候補を得る。

上記の方法でノード ID 候補が得られなかった場合は、すべての近隣ノードから式 (1) と式 (2) を使ってノード ID 候補を得る。

Construction of Overlay Network Considering Physical Network Topologies in MANETs

Yuuta Nishihara[†], Yuichi Suyama^{††}, Ueda Yuki^{††}, Yusuke Yokota[†] and Eiji Okubo[†]

[†]College of Information Science and Engineering, Ritsumeikan Univ.

^{††}Graduate School of Science and Engineering, Ritsumeikan Univ.

ノード ID の決定

上記の方法で得られたノード ID 候補の中から、ノード ID を決定する。ノード ID 候補が複数ある場合は、Successor・Predecessor ノードまでの経路が短いノード ID 候補を選択する。しかし、Successor・Predecessor ノードのハッシュ値間の距離が短いノード ID 候補は除外する。また、移動性が高いノードを Successor・Predecessor ノードにした場合、ノード ID の再計算が頻発する可能性があるため、移動性が高いノードが Successor・Predecessor ノードになるノード ID 候補も除外する。

ノード ID の再計算

ノードが移動すると、Successor・Predecessor ノードへの経路が長くなりネットワークへの負荷が増加してしまうため、ノード ID の再計算を行う必要がある。近隣ノードの状況や Successor・Predecessor ノードへの経路の変化を調べ、ノードが移動したことを検知して適切なノード ID の再計算を行う。しかし、この方法の場合、移動していないノードもノード ID を再計算してしまう可能性がある。そのため、ノードの位置情報やノードの移動を検知することのできるセンサが搭載されていれば、その情報を積極的に活用する。

3.2 経路情報の維持管理

Chord では、目的ノードを $\log(N)$ のホップで検索できるように、ハッシュ空間上で 2 の k 乗ずつ距離の離れたノードの表 (finger テーブル) を保持しており、ノードがネットワークに参加・離脱した際、自ノードを参照している $\log(N)$ 個のノードの finger テーブルを更新する必要がある。

しかし、MANET で finger テーブルの更新を徹底するとトラフィックが増大してしまう。そのため、 $\log(N)$ の探索効率をある程度諦めて、自ノードの Successor ノードか Predecessor ノードの保持している finger テーブルと経路情報を取得し流用する。本手法では、Successor・Predecessor ノードとの物理的な距離が短くなることが多いため、発生するトラフィックが少なく、流用できる情報が多い。

また、finger テーブルが必要になった時に更新したり、ノードを経由するパケットの経路情報を解析して更新するなどして消極的に行う。

3.3 レプリケーション

データ紛失対策の機構はインターネット上で DHT を使用したオーバーレイネットワークで提案されており、「定期的にデータの put(データの登録)を繰り返す」「DHT によって求められたハッシュ値周辺のノードにレプリカを配置する」「複数の DHT を使ってレプリカの配置位置を決定する」などがある。また、データにプライオリティを割り当て、そのプライオリティによってレプリカを配置する数を決めるといった仕組みがある。

MANET では、インターネットに比べてレプリケーションによるネットワークの負荷がより問題になってくる。そのため、慎重にデータのプライオリティを割り当てる必要がある。また、ノードの寿命が尽きそうなノードにレプリカを配置しても意味がなく、ネットワークの負荷が増大するだけである。Chord では通常、DHT の隣のノードにレプリカを配置するが、この時隣接ノードの寿命が尽きかけていた場合、さらに DHT の隣のノードにレプリカを配置するようにするという電力を考慮したレプリケーション手法も考えられる。

4 シミュレーション

本手法と Chord を、Network Simulator 2[3](以下、NS2 と記す) 上で実装し、シミュレーションを行った。本手法と Chord によって構築されるオーバーレイネットワークの、Successor ノードまでの経路長を比較し、本手法により Successor ノードまでの経路が短いオーバーレイネットワークが構築できることを確認する。以下、シミュレーション環境と結果について述べる。

4.1 シミュレーション環境

本シミュレーションで使用したパラメータの値を表 1 に示す。全てのノードは 20 秒間隔でオーバーレイネットワークに参加する。提案手法では、ノード ID の再計算の判断を 200 秒間隔で繰り返す。

表 1 シミュレーション条件

ノード数	50	75	100
フィールド (m × m)	900 × 900	1100 × 1100	1300 × 1300
移動モデル	Random Waypoint Model		
ノード移動速度	0~3m/s		
シミュレーション時間	2500s		

4.2 シミュレーション結果

シミュレーション結果を図 1 に示す。横軸はノード数で、縦軸は Successor ノードまでのホップ数の平均値である。図 1 から、Chord はノード数が増えるにしたがってホップ数が単調に増加しているが、提案手法はホップ数の増加を抑制できていることが解る。

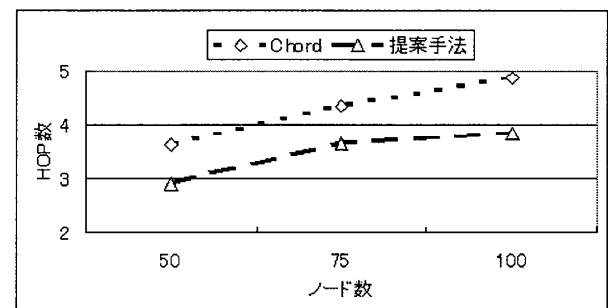


図 1 ネットワーク規模における Chord と提案手法の Successor ノードまでの平均ホップ数

5 おわりに

本稿では、MANET における物理ネットワークを考慮したオーバーレイネットワーク構築手法について述べた。今後の課題として、引き続きシミュレーションによる評価を行い、本手法の特性を明らかにする。

参考文献

- [1] T. Zahn, J. Schiller: "MADPastry: A DHT substrate for practically sized MANETs," In 5th Workshop on Applications and Services in Wireless Networks (ASWN), June 2005.
- [2] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan: "Chord: A scalable peer-to-peer lookup service for Internet applications," Proc. of ACM SIGCOMM 2001, pp 149-160. ACM Press, August 2001.
- [3] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/index.html>