

分散型ネットワーク座標系における座標決定の高速化 Acceleration of Coordinate Determination in Decentralized Network Coordinate System

木浦 正博[†] 廣安 知之[‡] 三木 光範[‡]
Masahiro KIURA Tomoyuki HIROYASU Mitsunori MIKI

1. はじめに

インターネットの普及にともない、インターネットを利用した様々なアプリケーションが開発されている。このようなアプリケーションにおいて、ユーザビリティとスケーラビリティを向上させるためには資源の複製や経路制御が重要となる。しかし、資源の複製や経路制御を行うためには、ノード間の遅延時間を測定することが必要であり、ネットワークに参加したノードすべてが遅延時間を測定することは非常に困難である。

ネットワーク座標系は、このような遅延時間を一部のノードとの実測によって実測していないノードの遅延時間を予測する。そのために、ネットワークに参加している各ノードを座標系に配置する。特に、Peer-to-Peerといった分散型ネットワークにおいては、正確かつ高速に各ノードのネットワーク距離を座標空間上に表現することが大きな課題である。

本研究では、ネットワーク座標系の一つである Vivaldi[1] について、そのパラメータを調整し、ネットワーク座標系における座標の修正を高速化する。これにより、実ネットワークをより早くネットワーク座標系に表現することが可能となる。

2. ネットワーク座標系

ネットワーク座標系は、ネットワークに接続したノード間の遅延時間を低次元座標空間に表現する。互いのノードは、ノード間の遅延時間をその座標空間内の距離として維持する。座標空間内で、ノードの位置関係表現するためには、遅延時間の実測が必要であるが、ネットワークに接続している少数のノードの遅延時間から、自身の座標を決定し、遅延時間を測定していないノードとの遅延時間は座標間距離によって予測する。

ネットワーク座標系は、それぞれのアルゴリズムから以下のように分類できる。

- ランドマーク型

ランドマーク型モデルでは、それぞれのノードは、自身の座標を決定するために、ランドマークノードを用いる。そのため、明確な相手ノードが決定している。一方、ランドマークノード以外のノードとは直接通信を行うことがないため、ランドマークノードに障害が生じた場合、座標が決定できないという欠点がある。

- シミュレーション型

シミュレーション型モデルでは、物理的なばねの伸縮を利用する。それぞれのノードは、力学的なベ

クトルや力を遅延時間を基にした式に置き換える。ランドマーク型と比べ、スケーラビリティが高いことが知られているが、それぞれのノードの座標は明確には決定されていない。

本研究では、もっとも広く利用されているシミュレーション型ネットワーク座標系アルゴリズムである Vivaldi に焦点を当て、座標の高速決定について考える。

3. Vivaldi

Vivaldi はシミュレーション型ネットワーク座標系の一つとして、Peer-to-Peer のように各ノードが分散的に座標変更を行うことで、ネットワーク上のノードの位置関係から、座標を決定し、予測することが目標である。前章で述べたように、明確なランドマークを用いることなく、少数のノードとの遅延時間を測定することによりバネモデルに基づいて自身の座標を決定することが可能である。Vivaldi において重要となるのは、ネットワークに参加している他のノードの座標を基に自身の座標を変更することである。

それぞれのノードの座標は自身以外のノードの座標に依存するため、他のノードの座標が大きく変化し続けている場合には、自身の座標も大きく変更し続けなければならない。座標が発振し、座標の収束に非常に時間がかかることが考えられる。Vivaldi では、この問題をエラーという指標を用いて改善している。具体的には、それぞれのノードはデータとして、自身の座標 x_i およびローカルエラー e_i をもち、他のノードの座標 x_j とそのリモートエラー e_j (自身以外のノードのローカルエラー) を利用し自身の座標を変更する。エラーはそのノードの座標間距離と遅延時間 r_{tt} の相対誤差として定義される。以上のことを表した Vivaldi アルゴリズムを以下に示す。

```
vivaldi( $r_{tt}, x_j, e_j$ ) {
  //ローカルエラーとリモートエラーを元に重みを計算
   $w = e_i / (e_i + e_j)$ 

  //相対誤差を計算
   $e_s = ||x_i - x_j|| - r_{tt} / r_{tt}$ 

  //重みと相対誤差を元にローカルエラーを更新する
   $e_i = e_s \times C_e \times w + e_i \times (1 - C_e \times w)$ 

  //座標を更新する
   $\delta = c_c \times w$ 
   $x_i = x_i + \delta \times (r_{tt} - ||x_i - x_j||) \times u(x_i - x_j)$ 
}
```

ノードの座標変更は、ローカルエラーおよびリモートエラーによる重み w とパラメータ c_c による δ によって

[†]同志社大学大学院 工学研究科

[‡]同志社大学 工学部

決定される。リモートエラーの小さいノードは、ほとんど座標変更を行わないため、そのノードの座標は座標変更の際に、遅延時間と座標間距離の差をできるだけ小さくするように移動する。一方、リモートエラーの大きいノードは、座標を大きく変更し続けているノードであるため、そのノードの座標を元にしてほとんど移動しない。

この Vivaldi のアルゴリズムについて次章では、本研究のテーマである座標の高速決定について述べる。

4. Vivaldi における座標決定の高速化

4.1 座標決定の高速化と座標発振の可能性

バネモデルをシミュレートする Vivaldi では、各ノードが自身以外の他のノードに依存し、自身の座標を高速に収束させた場合には、各ノードの座標の発散をまねく。絶対的座標指標を持たないシミュレーション型では、座標決定の高速化と座標発振の可能性は、トレードオフの関係にあると言える。そのため、 c_c は移動量を示すパラメータとして推奨値は 0.25 とされている。

4.2 重みによる移動量の調整

3章で示した Vivaldi アルゴリズムでは、それぞれのノードが座標を変更する際に e_j が高ければ、その相手ノードは座標変更において信頼性が低いノードであるとされた。すなわち、重み w はノードから見た他のノードの信頼性であると言え、この値を調節することで座標の収束率を上げることができる。しかし、前節で述べたように w を急激に変化させた場合、座標が収束すべき座標に到達しないことや座標が発振してしまう可能性がある。本研究では、重み w を算出する式を下記のように改良した。

$$k = e_j/e_i \quad (1)$$

$$w = 1/(1+k^n) \quad (2)$$

重み w はローカルエラーとリモートエラーによって算出される 0 から 1 の値である。そのため、 w 値を従来のアルゴリズムよりも減少させることで高速化をはかることができる。しかし、 w は座標決定において非常に重要な変数であるため、 w が大きく変化する改良を行うことはできない。そのため、今回は、パラメータ n を新たに設け、 n の調整に焦点を当てる。なお、(2) 式において、 $n = 1$ のとき、従来の Vivaldi と同じ働きとなる。この手法の有用性を実験により確かめる。

5. 評価実験

5.1 実験環境

本実験では、Vivaldi を用いて 400 ノードによるネットワークシミュレーションを行う。ノード間の遅延は、400 ノードが最小遅延時間 100ms とし、グリッド状に並ぶような遅延である。すでに 200 ノードは長方形格子状に並んでいるものとし、それらのローカルエラーの初期値は $e_i = 0.1$ 、残りの 200 ノードは初期点を原点とし、ローカルエラーの初期値は $e_i = 10$ とした。なお、各ノードは、自ノード以外のあらかじめランダムに選択した 6 ノードと通信を行う。

5.2 実験結果

シミュレーションによる実験の結果を図 1 に示す。図 1 では、ネットワークに参加しているすべてのノードペアの遅延時間と距離の差を用いて、累積分布関数によってグラフ化した。縦軸が累積分布関数値であり、横軸が遅延時間と距離の差である。

図 1 では、すべての遅延時間と距離の差の値について、 $n = 1.0$ すなわち通常の Vivaldi アルゴリズムによるグラフが、 $n = 1.4$ のグラフよりも下方に位置している。そのため、従来のアルゴリズムよりも高速化をはかることができたと言える。

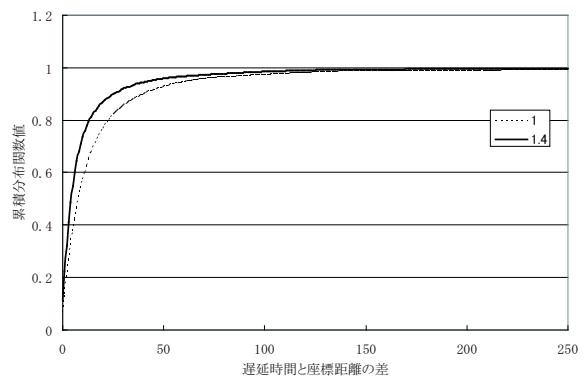


図 1: n の値による累積分布関数値の変化

6. 今後の課題

今回は、最小遅延時間を 100ms とし、グリッド状のネットワークのシミュレートを行った。その結果座標決定の高速化をはかることができたが、最小遅延時間が 100ms よりも大きい場合や 100ms よりも小さい場合、また Vivaldi を適用する個々の実ネットワークについても今後実験を行い提案手法の有効性を検証する必要がある。

7. まとめ

本研究では、シミュレーション型ネットワーク座標系アルゴリズムである Vivaldi について、最小遅延時間が 100ms のグリッド状のネットワークを想定し、そのパラメータを調整した。そして、従来の Vivaldi よりも高速かつ正確に座標が収束することを示した。今後は、Inet Topology Generator[2] や King dataset[3] を利用したシミュレーションを行うとともに、実ネットワークを用いた稼働実験を行うことでより改良を行う。

参考文献

- [1] F. Dabek, R. Cox, F. Kaashoek and R. Morris. "Vivaldi: A Decentralized Network Coordinate System," Proc. ACM SIGCOMM, pp.426-437, Aug. 2004
- [2] Inet Topology Generator, <http://topology.eecs.umich.edu/inet/>
- [3] kingdata, <http://pdos.csail.mit.edu/p2psim/kingdata/>