

アドホックネットワークにおける大規模な ネットワークでの運用可能なルーティングプロトコル

梅垣 正和^{†1} 安達 直世^{†2} 滝沢 泰久^{†2}

本研究ではノード数が増加しても高い配送率を達成するアドホックルーティングプロトコルを提案する。既存の多くのルーティングプロトコルはフラッディングを使うことによって最短経路を構築する。一方、提案するプロトコルはフラッディングを使わず P2P をネットワークレイヤに適用することで経路を構築する。これにより、ネットワーク全体でのルーティング情報の構築・維持に消費するパケットの数を減少させることによって、高い配送率を達成したことを示す。Qualnet 4.5 によるシミュレーションでは他のプロトコルと比較して提案プロトコルの配送率の高さを示して検証を行う。

Scalable Routing Protocol in Large Scale Ad-hoc Network

MASAKAZU UMEGAKI,^{†1} NAOTOSHI ADACHI^{†2}
and YASUHISA TAKIZAWA^{†2}

In this paper, we propose a ad-hoc routing protocol that achieves a high delivery rate even if the number of nodes is large. A lot of existing routing protocols construct the shortest path by using the flooding. On the other hand, proposed protocol constructs the route with P2P routing concept to network layer and without flooding. As result, it is shown a high delivery rate by decreasing the number of packets for construction and maintenance of routing information on the entire network. The simulation result by Qualnet 4.5 shows that proposed protocol achieves high delivery rate against other ad-hoc routing protocols.

1. はじめに

近年、アドホックネットワークに関する研究が盛んである。アドホックネットワークは既存のインフラストラクチャが不要で自律的にネットワークの構築が可能であるが、既存のワイヤレスネットワークとは別の問題が生じてくる。その中のひとつにルーティングの問題がある。アドホックネットワークでは端末が目的の端末と通信を行う場合、目的の端末までの経路を探索する必要がある。既存のルーティングアルゴリズムの多くは目的の端末までの経路をフラッディングにより探索を行う。フラッディングの手法を用いるルーティングアルゴリズムの多くは、目的のノードへの経路は最短経路を選択することが可能である。

しかし、フラッディングを用いる手法はネットワーク全体でのノード数が少ない場合は問題ないが、ネットワーク全体でのノード数が多いとフラッディングによる制御パケットがネットワーク全体に拡散してしまい、帯域幅の圧迫やパケットの衝突を起こす。その結果、最適な経路を構築・維持することが困難になるため、データパケットが宛先のノードまで届く率(配送率)が低くなり、ノード数の増加に対するスケーラビリティを維持できなくなる¹⁾(Broadcast Storm 問題と呼ばれる²⁾)。また、フラッディングを使わない方法として GPS を用いる方法がある。しかし、ノードを大量に散布する場合はコスト面で不利であるため適切ではない。

近年ではインターネットにおける P2P のオーバーレイネットワーク (Overlay Network) と呼ばれるアーキテクチャをアドホックネットワークに適用しているプロトコルが複数提案されている⁷⁾。インターネットでの P2P ネットワークとアドホックネットワークは、どちらもネットワークへの参加と離脱、自立組織化、分散化という概念を共有しているため、P2P ネットワークのアルゴリズムをアドホックネットワークに適用することができる⁸⁾。最近では VRR¹⁾, SSR⁹⁾, MADPastry¹⁰⁾, MeshChord¹¹⁾ 等が P2P を応用した研究事例である。しかしながら、これらの P2P を応用したプロトコルはルーティングテーブルの構築の際にフラッディングを用いているため、経路探索による帯域の消費が大きい。

そこで提案手法では P2P をネットワークレイヤに適用してフラッディングを一切使用し

†1 関西大学大学院 理工学研究科 〒 564-8680 大阪府吹田市山手町 3-3-35
Graduate School of Engineering, Kansai University 3-3-35 Yamate-cho, Suita-shi, Osaka, 564-8680 Japan

†2 関西大学 環境都市工学部 〒 564-8680 大阪府吹田市山手町 3-3-35
Department of Civil, Environmental and Applied System Engineering, Faculty of Environmental and Urban Engineering, Kansai University 3-3-35 Yamate-cho, Suita-shi, Osaka, 564-8680 Japan

ないことで、経路の構築や維持に掛かる帯域消費を減らし、ノードの数が増えた場合でも高い配送率を達成してスケーラビリティが向上するルーティングプロトコルを提案する。

2. 既存のルーティングプロトコルと研究の動向

アドホックワイヤレスネットワークのルーティングプロトコルは構造的な特徴から複数の分類に分けられる。それはフラット型、階層型、位置情報補助型、複数経路型などである。フラット型は大きく分けて、リアクティブ (Reactive) 型プロトコルとプロアクティブ (Pro-Active) 型プロトコル、リアクティブ型プロトコルとプロアクティブ型プロトコルを組み合わせたハイブリッド (Both Pro-Active and Reactive) 型プロトコル、および P2P を利用したグループに分けられる。

2.1 リアクティブ型ルーティングプロトコル

リアクティブ型ルーティングプロトコルは通信要求があったときにルーティングテーブルを構築する。代表的な方式として DSR³⁾, AODV⁴⁾ がある。ノードが宛先までの経路が必要になったときにネットワーク内をフラッディングで宛先までの経路を探索する。この探索は経路が見つかるか、すべての経路を探索すると終了する。見つかった経路は宛先までの経路が壊れるまで維持される。

2.2 プロアクティブ型ルーティングプロトコル

プロアクティブ型ルーティングプロトコルは通信要求がなくてもあらかじめルーティングテーブルを構築しておく。したがって、通信要求が発生すると直ちに通信を開始できる。また、ルーティングテーブルを常に最新の状態に維持するため、定期的な情報の交換やネットワークのトポロジーに変化があるとネットワーク全体に変更があった経路の更新情報を伝える。そのためプロアクティブ型のプロトコルでは基本的にすべてのノードがネットワーク全体のトポロジーを把握しており、トポロジーから最短経路を Dijkstra のアルゴリズムを使って求めるプロトコルが多い。代表的な方式として DSDV⁵⁾, OLSR⁶⁾ がある。

2.3 P2P (Peer-to-Peer) の応用

P2P を応用したアドホックルーティングプロトコルは主に透過型、クロスレイヤー型、統合型の三種類に分けられる。透過型は既存のルーティングプロトコル上に P2P のレイヤーを実装したプロトコルである。これはインターネットの P2P と同じであるため、物理レイヤーがアドホックネットワークであることを考慮していないため、パフォーマンスが悪い。それを改善したのが、クロスレイヤー型と統合型である。しかしながら、これらの P2P を応用したプロトコルはルーティングテーブルの構築の際にフラッディングを用いているた

め、経路探索による帯域の消費が大きい。また、図 1 のオーバーレイにおけるトポロジを構築する際、物理上のトポロジーを考慮していないため、近くに目的のノードが存在するのにも関わらず、迂回してしまうといった問題がある。

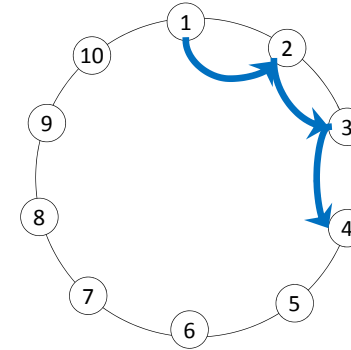


図 1 オーバーレイ上の仮想リング

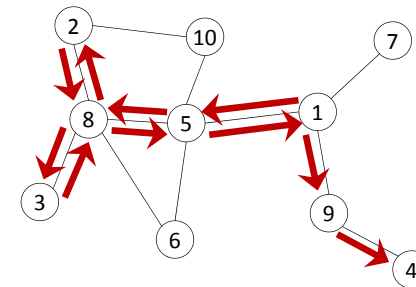


図 2 物理的なトポロジ

3. 提案プロトコル

既存のルーティングプロトコルではフラッディングを用いることにより、経路探索のパケットがネットワーク内に溢れてしまい、帯域幅を圧迫やパケットの衝突を繰り返すことで配送率の低下を招いてしまう。これが既存のルーティングプロトコルにおいてスケラビリティを達成できない原因である。そのためフラッディングを使用しないことで、ネットワーク内の経路探索に掛かる帯域消費を減らすことで配送率に対してスケラビリティ向上を狙う。提案するプロトコルはリアクティブ型プロトコルとプロアクティブ型プロトコルにP2Pをネットワークレイヤーに応用した統合型のハイブリッドプロトコルであり、ノード数が多い場合でも高い配送率を可能とする。

提案プロトコルは以下の二つのフェーズを持つ。

- ルーティングテーブルを構築するフェーズ
- アプリケーションのデータパケットを転送するフェーズ

以下、それぞれのフェーズについて説明を行なう。

3.1 ルーティングテーブルの構築フェーズ

ルーティングテーブルを構築するフェーズはさらに二つのプロセスを持つ。

- (1) ノードはネットワークに参加すると Neighbor と双方向のリンクを確立する。ここで Neighbor とは電波の通信範囲内にいる直接通信可能なノードである。また、提案プロトコルは一对多のブロードキャスト送信ではなく、一对一のユニキャスト通信を主に使用する。そのためパケットが失われると情報がそこで途切れてしまうため、安定したリンクが必要である。そこで、双方向に通信が可能であることを確認する過程で一回でもメッセージが失われた場合は、不安定なリンクとして Neighbor には登録を行わない。

ここで仮想的なパスを次のように定義する。仮想的なパスとはオーバーレイ上に存在するノードへの経路である。例えば、図1の矢印が仮想的なパスであり、図2の矢印が仮想的なパスに対する物理的なパスである。このようにオーバーレイ上に存在する全てのノードが隣接しているノード(自ノードID \pm 1)へのパスを構築することによって、仮想的なパスを順々に辿ることで全てのノードに到達できる。しかしながら、図2に示すように単に順々に辿るだけでは目的のノードが近くに存在するにも関わらず迂回するような経路が構成されてしまう可能性がある。そこで次にオーバーレイ上で仮想的なショートカットを構築する。

- (2) 仮想的なショートカットとはオーバーレイ上で隣接したノードを越えるパスである。仮想的なショートカットはネットワーク全体での最小のノードID \leq 自ノードID $\pm 2^n$ (n はノード数) \leq ネットワーク全体での最大のノードID に該当するIDを探索することで構築する。このように探索するノードIDの数値を指数的に増やして、オーバーレイ上で自ノードIDに近いIDを密にして遠いIDを疎にすることで、ネットワーク全体のノード数に対するショートカットの構築に消費するパケット数を抑える。また、全てのノードはネットワーク全体での最大のノードIDと最小のノードIDを既知とする。探索はパケットがネットワークに拡散しない一对一のユニキャスト通信で行い、自身のルーティングテーブルの中から探索IDに絶対値の差が最も小さいIDを知っている Neighbor に転送を繰り返すことにより、物理的なパスを構築する。ここから具体的な例を用いて説明を行う。探索するノードIDの例を表1に示す。ノード数が50、自ノードIDが10とすると、上記の探索に該当するIDは $10 - 2^3 = 2, 10 - 2^2 = 6, 10 - 2^1 = 8, 10 - 2^0 = 9, 10 + 2^1 = 11, 10 + 2^2 = 14, 10 + 2^3 = 18, 10 + 2^4 = 26, 10 + 2^5 = 42$ である。これを絶対値の差が大きい順に並び替えると、順番は表1ようになる。また、必ずしも探索したノードIDを見つけないといけないということではない。あくまでもショートカットであり、図3のようにオーバーレイ上で自ノードIDから離れたノードIDへのパスを構築することで、ネットワークレイヤにおける物理的なホップ数の減少を図ることが目的である。これはノードがランダムに配置されているため、任意のノード間のホップ数を平均値(期待値)として仮定できる。そのため、オーバーレイ上で中継するノードが減れば、ネットワークレイヤでの物理的なホップ数も削減できることが期待できる。探索を行なう順番は自ノードIDから最も離れたIDから行なう。こうすることで自ノードIDに近いIDを探索するまでに、認知できるノードの数が増やすことで自ノードIDに近いIDへのパスが構築できることを期待する。これは仮想リングを順々に辿っていくことで全てのノードへの到達が保証されているため、自ノードIDに近いIDへの経路が増えるほど任意のノードへ到達できる可能性が上がるためである。このことから、自ノードID $\pm 2^n$ を探索することはオーバーレイ上で自ノードIDに近いIDへの探索を密に行なうため適している。

3.2 アプリケーションデータパケット転送フェーズ

- (1) アプリケーションからデータパケットの送信要求があるとノードは自身のルーティングテーブルを参照する。

- (2) 宛先のノード ID がルーティングテーブルに存在する場合は、データパケットの送信を直ちに開始する。
- (3) 宛先のノード ID がルーティングテーブルに存在しない場合は、ルーティングテーブルの中から絶対値の差が最も小さいノード ID へ転送を繰り返し行ことで宛先を見つける。
- (4) 宛先を見つけたら探索情報を元に、データパケットを自ノードの通信可能な範囲内でより遠くに存在するノードに転送することでホップ数を少なくする。

このように提案プロトコルはルーティングテーブルの構築からデータパケットの送信まで、フラッディングを一度も使用しない。そのためネットワーク内に無駄なパケットが流れることがなく、ネットワーク全体の帯域消費を下げている。

表 1 探索の順番の例 (ノード数 50, 自ノード ID:10)

探索する順番	42 ⇒ 26 ⇒ 18 ⇒ 2 ⇒ 14 ⇒ 6 ⇒ 12 ⇒ 8 ⇒ 11 ⇒ 9
--------	---------------------------------------------

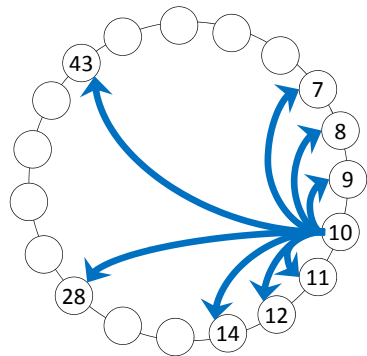


図 3 オーバーレイ上でのショートカット

図 4 はノード ID:10 のルーティングテーブルを視覚的に表したのものである。このルーティングテーブルはネットワークレイヤのトポロジー情報を表したものである。この例では Neighbor は三つある。そして、Neighbor に接続してあるノード群は Neighbor を通じて認知

したノード ID である。このようにルーティングテーブルは直接通信可能な Neighbor と、その Neighbor を通じて認知することができたノード群から構成されている。また、Neighbor を通じて認知することができたノードは、オーバーレイ上で仮想的なショートカットを構築するために、探索を行ない収集したノードである。そのため、ノードがネットワークから離脱したり、パスが壊れない限りは、ルーティングテーブルに存在しているノード ID との通信が保証されている。

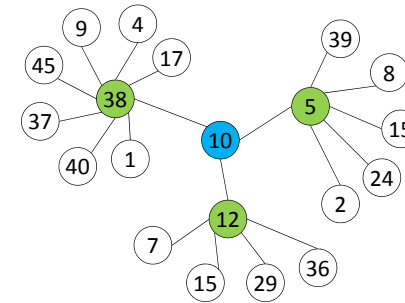


図 4 ルーティングテーブルを視覚的に表したものの

4. シミュレーションと検証

提案するプロトコルの性能を評価・検証するために、ネットワークシミュレータ Qualnet 4.5 に実装してシミュレーションを行った。シミュレーションを実行したコンピュータの環境を以下に示す。OS:Ubuntu 10.04 LTS, CPU:Intel Xeon E5530, Memory:24GByte。また、シミュレーションの条件は以下の通りである。ノードの数は 100, 200, 300, 400, 500 の五種類で、ノードの配置はランダムである。また、乱数の Seed 値を変えることにより各十種類のトポロジーを用意した。ノードの通信半径は約 300m で、対象の領域はノードの密度が一定にして、Neighbor の数が大体 10 個になるようにしている。ノード数が 100 のときは 1900 × 1900, ノード数が 200 のときは 2700 × 2700, ノード数が 300 のときは 3300 × 3300, ノード数が 400 のときは 3800 × 3800, ノード数が 500 のときは 4200 × 4200 である。ワイヤレスデバイス は IEEE 802.11b であるが、帯域幅を 2Mbps に制限している。次にシミュ

レーションのシナリオを以下に示す。シミュレーションを開始してからルーティングテーブルを構築し終えるまで 100 秒間待機する。ノードの数によりルーティングテーブルが構築し終える時間は異なるが、概ね 100 秒間待つことでどのノード数でもルーティングテーブルは完成している。次に各ノードはランダムに宛先を選択して 512byte のデータパケットを一斉に送信する。そして 60 秒後に各ノードは再度ランダムに宛先を選択して 512byte のデータパケットを一斉に流す。前述をシミュレーションが終わる 3600 秒まで繰り返し行う。以上の条件を表 2, 3, 4 に示す。提案するプロトコルと比較するのはリアクティブ型プロトコルの DSR と AODV, プロアクティブ型プロトコルの OLSR である。このシナリオは各ノードが一斉にデータパケットを送信して帯域の圧迫とパケットの衝突を意図的に起こす。これにより提案するプロトコルが経路の維持や探索に掛かる帯域消費を減らし、配送率を上げることに成功したことを示す。

表 2 シミュレーション実行の環境

Simulation Soft	OS	CPU	Memory
Qualnet 4.5	Ubuntu 10.04 LTS	Intel Xeon E5530	24GByte

表 3 シミュレーションの環境

Number of Nodes	Field(m)	Wireless Device
100	1900 × 1900	
200	2700 × 2700	
300	3300 × 3300	IEEE 802.11b(MAX:2Mbps)
400	3800 × 3800	
500	4200 × 4200	

表 4 シナリオの設定

Src	Dest	Packet	Interval	Simulation Time
Random	Random	512byte	60sec	3600sec

結果を図 5, 6, 7 に示す。図において、メモリ不足などでシミュレーションを完走できなかった箇所はプロットしていない。

図 5 は提案プロトコルと DSR, AODV, OLSR と配送率を比較した結果である。AODV と OLSR は共に非常に低い配送率である。一方、提案プロトコルと DSR はノード数が 500 の時点でも 80% 付近である。比較対象のプロトコルはフラッディングによるパケットの衝突や再送制御等で大量のイベントが発生しているため、経路の構築・維持が困難になったことが配送率の低下に繋がっている。同じリアクティブ型ルーティングプロトコルである DSR と AODV に差が開いている。これはシナリオに移動がないため、AODV よりも DSR のほうが経路のキャッシュを積極的に行うためであると考えられる。提案プロトコルはフラッディングを使用しないことでネットワーク内に流れる無駄なパケットを削減したことが高い配送率に繋がっている。図 6 は提案プロトコルと DSR, AODV, OLSR と遅延を比較した結果である。ノード数が 400 のときに AODV が突出している以外は、提案プロトコルの遅延が大きい。これは提案プロトコルが最短経路を選択せずに、迂回するような経路を選択しているからである。図 7 は、ブロードキャストとユニキャストを合わせた全ての送信パケット数と、受信したアプリケーションのパケット数の比率を提案プロトコルと DSR で比較した結果である。AODV と OLSR は到達率が低かったため、計測は行っていない。尚、算出するにあたって、パケットの総数に再送のパケットを加えてある。提案プロトコルは DSR とほぼ同じ結果である。これは提案プロトコルのルーティングテーブルの構築と迂回するような経路によるパケットの消費が DSR のフラッディングによる経路の構築と最短経路によるパケットの消費とほぼ同じであることを示している。

5. まとめ

本研究ではフラッディングを使用しないことでネットワーク内に流れる無駄なメッセージを減らした。また、オーバーレイ上での仮想的なショートカットを構築することによりホップ数の削減を行った。その結果、提案するプロトコルは高い配送率を達成した。課題としては、今回提案したプロトコルは移動のことまで考慮されていない。アドホックネットワークでは端末が移動するため考慮する必要がある。

参考文献

- 1) M. Caesar, M. Castro, E. B. Nightingale, and G. OShea, "Virtual ring routing: Network routing inspired by DHTs," In *Proc. of ACM SIGCOMM*, Pisa, Italy,

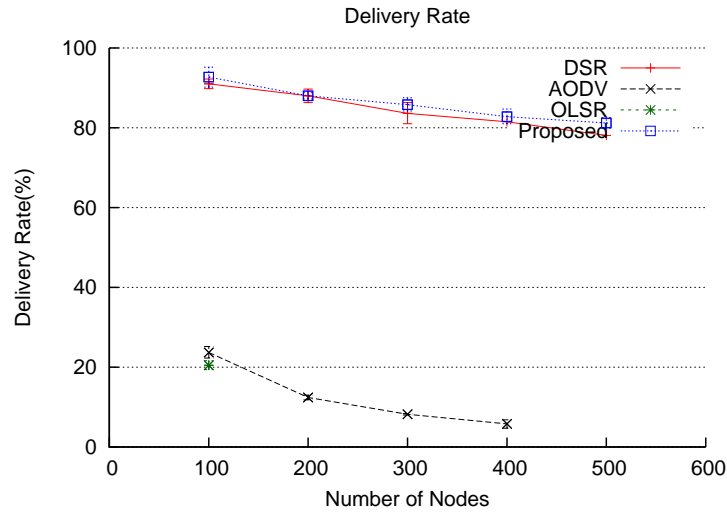


図 5 配送率の比較

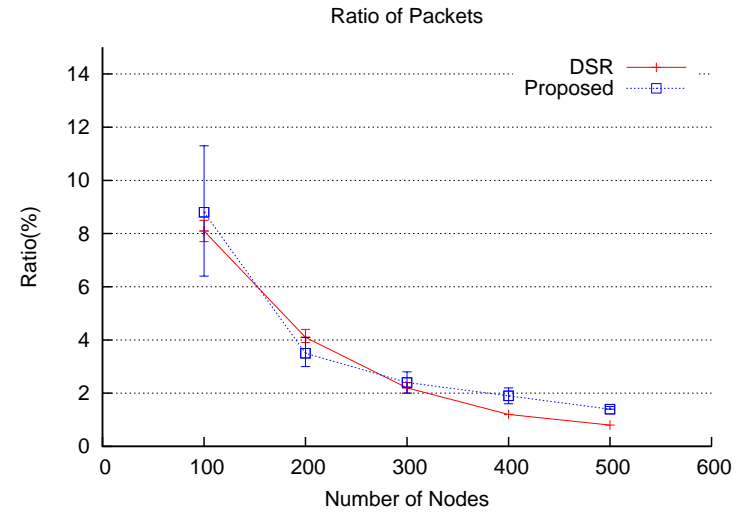


図 7 送信パケットと受信した App パケットの比率

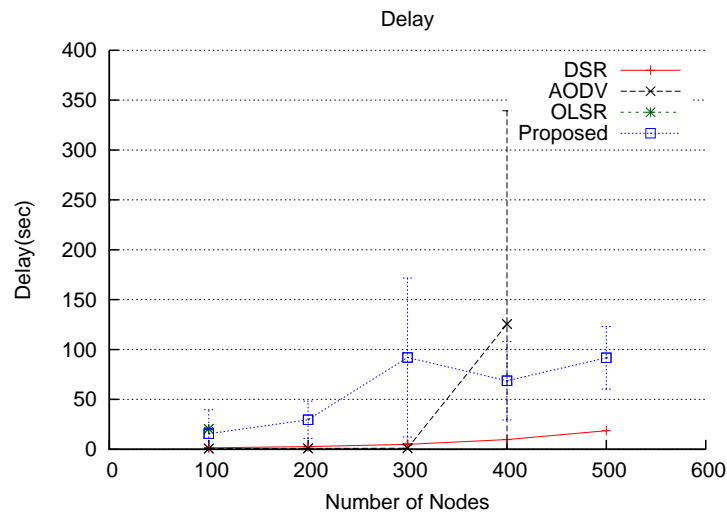


図 6 遅延の比較

September 2006.

- 2) 阪田史郎, "ユビキタス技術 センサネットワーク", オーム社, 2006
- 3) D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," In Ad Hoc Networking, 2001.
- 4) C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," In Mobile Computing Systems and Applications, February 1999.
- 5) C. P. P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," In Proceedings of ACM SIGCOMM '94, pp. 234-244, September 1994.
- 6) T. Clausen, P. Jacquet and Project Hipercom, "Optimized Link State Routing Protocol (OLSR)," IETF Request For Comment (3626), October 2003.
- 7) M. Castro, A. Kessler, C. Chiasserini, C. Casetti, and I. Korpeoglu, "Peer-to-peer overlay in mobile ad-hoc networks," *Handbook of Peer-to-Peer Networking*. 2010.
- 8) Y. Charlie Hu, Saumitra M. Das, and Himabindu Pucha, "Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks." CRC Press, 2005.
- 9) T. Fuhrmann, P. Di, K. Kutzner, and C. Cramer, "Pushing chord into the underlay: Scalable routing for hybrid manets," Technical report, Universitt Karlsruhe

(TH), June 2006.

- 10) T. Zahn, "Structured Peer-to-Peer Services for Mobile Ad Hoc Networks," Phd thesis, Freien University Berlin, 2006.
- 11) S. Buresi, C. Canali, M. E. Renda, and P. Santi, "Meshchord: A location-aware, cross-layer specialization of chord for wireless mesh networks," In *Proc. of PerCom*, Hong Kong, pages 206-212, 2008.