

MANETにおけるTCPスループットを考慮したルーティングアルゴリズムの提案と評価

M2015SC019 山下瑞樹

指導教員 河野浩之

1 はじめに

今までMANETと呼ばれるアクセスポイントを必要としない機器間無線通信ネットワークを構築する研究は多くなされている[3][4]。通常、MANETにおける経路選択手法は最短経路を選択するようになっている。しかし、地形や障害物による無線の減衰、ノードの密度、通信性能による電波品質の良し悪しによって、必ずしも最短経路を選択することが最善の経路であるとはいえない。また、ノードによって様々な方向や速度で移動するため急激なトポロジ変化を引き起こし、安定したネットワークを維持し続けることを困難にしている。これらの問題に対してMANETをクラスタリングすることによるフラッディングの効率化、トラフィック状況を基に評価値を設定することによる最善な経路選択手法を提案した研究があるが、どれもUDP通信によるものでTCP通信を考慮した研究は非常に少ない。現在モバイル利用者におけるトラフィックの94%がTCP通信であることを考えると、MANET上においても同様の利用が想定される。

本研究では、TCPスループットを考慮したメトリクスを導入することにより、実世界で利用されると想定された環境下においてネットワークの安定化と純粋なTCP通信のためのスループット向上を図る。また、MANETより利用が想定される状況から通信頻度とバッテリー消費の少ないリアクティブ型プロトコルであるDynamic Source Routing(DSR)を採用する。我々は、これを本研究用に改良することにより目的を実現する。実験には、ネットワークシミュレータ3.25(ns-3)を用いてMANETにおけるモビリティシナリオを構築し、提案するルーティングアルゴリズムの追加モジュールをC++で実装し、性能を評価する。

本稿は、全5章構成となっている。2章では、MANETにおける諸問題を解決するために提案されたクラスタリングや経路選択アルゴリズムについて紹介する。3章では、本稿で提案するMANET環境下におけるTCPスループットを考慮したルーティングアルゴリズムについて説明する。4章では、提案したルーティングアルゴリズムをns-3により構築した環境化でシミュレートし、既存のルーティングアルゴリズムと比較・評価する。5章では、本研究の総括を述べる。

2 関連研究

この章では、本研究で提案するアルゴリズムに用いるルーティング・プロトコルと関連する研究について説明する。

2.1 DSR

DSRは、IETF MANET WGで提案された経路制御プロトコルの一つである。DSRはその中でもリアクティブ型のルーティングプロトコルであり、経路要求がされて初めて送信元ノードから宛先ノードまでの経路を確立する。通信に関しては、送信パケットのヘッダ内のオプション(Route Request)に送信元ノードから宛先ノードまでのIPアドレスを指定するソースルーティング方式に従う。各ノードはまわりのノードと通信するたびに、宛先ノードまでの経路をRoute Cacheとして保持する。もし、Route Cacheに宛先ノードまでの経路が存在していない場合、送信元ノードは経路要求をブロードキャストする。宛先ノードまでの経路が複数存在した場合は、ホップ数の最も少ない経路を選択する。宛先ノードまたは宛先ノードまでの経路をキャッシュしているノードが経路要求に対する応答(Route Reply)が返されなければ、Route Cacheからその経路を破棄する。ノード間のパケット転送の数に比例し、ヘッダ情報も多くなりオーバーヘッドも増加するため、数十ノードを越える長距離通信は考慮されていない。

2.2 クラスタポイントを用いたクラスタリング [4]

鳴海らは、高移動・高通信頻度環境化において各クラスタにクラスタの中心点(クラスタポイント)と評価値を設け、常に最適なCluster Head(CH)を選出することで、安定的なクラスタの形成と維持を実現させた。効率的なフラッディングを実現するためには、各CHはクラスタの中心位置にすることが望ましい。しかし、動的な環境においてCHが常にクラスタの中心に滞在することは困難であるため、評価式(1)を用いてCHを動的に選出する。 n はクラスタ数、 v_{avg} はノードの平均移動速度、 d はクラスタポイントとの距離、 C は定数である。

$$E_n = C \times d + \sqrt{v_{avg}} \log_{10}(v_{avg} + 1) \quad (1)$$

CHからのビーコンを受信したノードは、クラスタポイントに対する評価値を算出し、評価値の最も低いノードを新たなCHに選出する。旧CHは新CHからビーコンを受信すると旧CHが保持していたノードリストを新CHへ送信しCluster Member(CM)となる。この一連の動作を定期的に行うことで高移動・高通信頻度環境下においても安定したクラスタの維持ができる。

2.3 優先度メトリクスに基づくルーティングアルゴリズム [3]

浅沼らは、各ノード間のトラフィック量を考慮したルーティングアルゴリズムを提案したことにより、高移動型

ノードにおいて安定した通信の実現と動的に経路を入れ替えることによりネットワーク全体のスループットの向上を実現した。DSR をベースとしたクラスタベースルーティングは単一経路のみを使用するに対し、提案手法は通常の経路（プライマリ経路）の他に代替経路（セカンダリ経路）を1本以上用意する。経路を複数用意することで経路内に別の通信トラフィックが発生し、輻輳が生じた場合にも送信元ノードから宛先ノードまでの通信経路のホップ数、トラフィック量、すべての隣接ノードとのリンク状態における平均値、バッテリー残量を総合的に判断し、プライマリ経路とセカンダリ経路を比較することで常に最適な経路を選択することができるようになり、ネットワーク全体の安定化とスループットの向上を図る。以下の評価式 (2) により優先度マトリクスを用いて評価値を算出することができる。

$$W = \omega_1 \times \frac{1}{N} + \omega_2 \times \frac{1}{traffic} + \omega_3 \times \frac{1}{LLT} + \omega_4 \times \frac{1}{B} \quad (2)$$

3 提案アルゴリズム

この章では、本研究で提案するルーティングアルゴリズムについて述べる。

3.1 提案手法

本研究は、動的なノードに対して有効的な手法であるクラスタリングを用いた DSR を拡張し、TCP スループットを考慮した複数のメトリックによる評価式を適用することで TCP 通信のための最適な経路を選択するルーティングアルゴリズムを提案する。一般的に TCP スループットは、送信元ノードから宛先ノードまでの通信経路のホップ数、隣接ノードとの往復遅延時間、パケット損失率、経路上のトラフィック量、ノードの移動速度、電波品質などが影響を与える要素として考えられる。特に、遅延が少ないこと、パケット損失率が低いことが TCP スループットの増加に寄与する。そのため本研究では、これらのメトリックを用いることにより、効率的で安定的な TCP 通信とノードのバッテリー消費を制限し、大規模な MANET 環境下においても通信動作の向上を図る。

3.2 TCP スループット算出モデル

TCP スループットを算出するモデルを用いることで、TCP スループットが最も高い経路を選択することが可能になる。TCP スループットを算出するモデルとして一般的に有名なものは、Mathis らが提案したモデルに改良を加えた Floyd らのモデル [1] と Padhye らが提案したモデル [2] が存在する。

Floyd らによるモデルを以下に示す。

$$T \leq \frac{1.5\sqrt{\frac{2}{3}} \times B}{RTT \times \sqrt{P}} \quad (3)$$

式 (3) の Floyd らによるモデルは、TCP の最大セグメントサイズ B 、経路上の最小遅延 RTT (Round Trip Time)、

送信元ノードから宛先ノードまでのパケット喪失率 P を計算し、TCP スループットの推定値 T を算出する。

Padhye らによるモデルを以下に示す。

$$T \approx \frac{1}{RTT \sqrt{\frac{2bP}{3}} + T_0 \min(1, 3\sqrt{\frac{3bP}{8}}) P(1 + 32P^2)} \quad (4)$$

式 (4) の Padhye らによるモデルは、経路上の平均遅延 RTT 、1 つあたりの ACK によって確認できるパケット数 b 、送信元ノードから宛先ノードまでのパケット喪失率 P 、タイムアウトの発生時間 T_0 を計算し、TCP スループットの推定値 T を算出する。

以上のモデルを基に TCP スループットを算出した結果、Floyd らによるモデルと Padhye らによるモデルは以下の表 1、表 2 となった。なお、式 (4) のパラメータの設定値は、通常の TCP に従って b を 2、 T_0 を $4RTT$ 、セグメントサイズを 1500byte としている。

表 1 Floyd らによる TCP スループット算出モデル

喪失率 \ 遅延時間	10ms	20ms	30ms	40ms	50ms
0.01%	1837kbps	919kbps	612kbps	459kbps	367kbps
0.02%	1300kbps	650kbps	433kbps	325kbps	260kbps
0.03%	1061kbps	530kbps	354kbps	265kbps	212kbps

表 2 Padhye らによる TCP スループット算出モデル

喪失率 \ 遅延時間	10ms	20ms	30ms	40ms	50ms
0.01%	1825kbps	913kbps	608kbps	451kbps	358kbps
0.02%	1284kbps	642kbps	428kbps	317kbps	244kbps
0.03%	1043kbps	521kbps	348kbps	260kbps	202kbps

本研究で用いる TCP スループットを考慮した経路選択は、複数ある経路の中から TCP スループットが高い経路を優先的に選択するルーティングアルゴリズムである。そのため、この経路選択においては正確な TCP スループットを算出する必要はなく、あくまで複数ある経路の中で TCP スループットの優劣が判定できれば良い。TCP スループットの指標として表 1、表 2 を比較した際、両モデルによる数値からグラフの線形が酷似していることがわかる。そのため、両モデルによる TCP スループットの推定の正確性も同等であることがわかる。また、Floyd らの研究は Padhye らの研究を参考にしている。その結果、本研究で用いる TCP スループット算出モデルは、計算が比較的簡単である Floyd らのモデルとした。

3.3 RTT&RTPL 算出モジュール

Floyd らの式 (3) より TCP スループットの推定値を算出するためには、送信元ノードから宛先ノードまでの RTT と往復パケット喪失率 ($RTPL^1$) が必要になる。特に、このメトリクスを取得するにあたって通信経路は、往路と復路とも同一でなければならない。そのため、送信元ノードは宛先までの全ての経路を指定するソースルーティング方式である DSR を採用する。送信元ノードが TCP スループットの推定値の算出に必要な各種メトリクスを取得する手順を以下に説明し、図 1 に示す。

¹Round Trip Packet Loss ratio

- (1) 各ノードは、起動したときから今までに転送したパケットからリンク毎にパケット喪失率を計算する．送信元ノードは宛先ノードに向けて通信を始めると、送信元ノードから宛先ノードまでの RTT と RTPL を要求するパケット (RTTPL² 要求) を一定間隔で送信する．
- (2) 送信元ノードと宛先ノードの中間経路に存在するノードは、RTTPL 要求を受信すると各リンクとのパケット喪失率を RTTPL 要求に付加して、宛先ノードへ転送する．
- (3) 宛先ノードが RTTPL 要求を受信したら各リンクのパケット喪失率から RTPL を計算し、RTTPL 応答を送信元ノードに送信する．
- (4) 送信元ノードが RTTPL 応答を受信したら、RTTPL 応答から RTPL を取得し、RTT は RTTPL 要求を送信してから RTTPL 応答を受信するまでにかかった時間を計算することで求める．

以上の手順により入手した RTT と RTPL を Floyd らの式 (3) に代入することで TCP スループットの推定値を算出する．

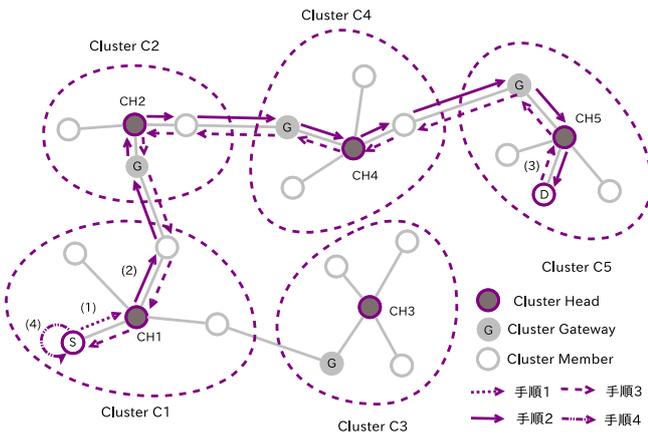


図 1 TCP スループット算出手順例

RTT 算出手法

送信元ノードは、RTTPL 要求を送信する際、そのパケットへタイムスタンプを付加する．宛先ノードは、RTTPL 要求のタイムスタンプを RTTPL 応答に入れ送信元ノードに送信する．そして、送信元ノードが RTTPL 応答を受信した時刻と RTTPL 応答のタイムスタンプの差を RTT とする．

RTPL 算出手法

各ノードは隣接ノードとのリンク状態からパケット喪失率を測定する． $N_1, N_2, \dots, N_i, \dots, N_{n-1}, N_n$ 個のノードからなる経路において、 N_i から N_{i+1} へ向かうリンクのパケット喪失率を $P_{i,i+1}$ とすると、 i 番目のノードである N_i は N_i から N_{i-1} へのパケット喪失率 $P_{i,i-1}$ と N_i から N_{i+1} へのパケット喪失率 $P_{i,i+1}$ を管理する． N_1 が

RTTPL 要求を N_n へ向けて送信し、中間ノードである N_i が RTTPL 要求を受信した場合、 N_i は $P_{i,i-1}$ と $P_{i,i+1}$ を RTTPL 要求のパケットに格納し、次のノードである N_{i+1} へ転送する．よって、宛先ノード N_n が RTTPL 要求を受信したときパケットに含まれるパケット喪失率は全部で、 $P_{1,2}, P_{2,3}, \dots, P_{n-1,n}, P_{n,n-1}, \dots, P_{3,2}, P_{2,1}$ となる．ちなみに、 N_n は式 (5) から算出した RTPL を RTTPL 応答へ格納し、 N_1 へ送信する．

$$RTPL = 1 - \prod_{i=1}^{n-1} (1 - P_{i,i+1})(1 - P_{i+1,i}) \quad (5)$$

ここで、もし送信元ノードが RTTPL 要求を送信したにも関わらず、宛先ノードから RTTPL 応答が返ってこない場合、RTTPL 要求もしくは RTTPL 応答が途中の経路上で喪失したことになる．その場合は、改めて送信元ノードが RTTPL 要求を送信し、RTPL を算出する．

3.4 TCP スループットをメトリクスに加えた経路選択機構

後述する 3.3, 3.4 節で測定したメトリクスを用いて送信元ノードは最終的に経路を決定する優先度 W を算出する．以下に優先度 W を算出する式 (6) を示す．

$$W = \omega_1 \times \frac{1}{N} + \omega_2 \times \frac{1}{T} + \omega_3 \times B \quad (6)$$

ここで、 N は送信元ノードから宛先ノードまでのホップ数、 T は送信元ノードから宛先ノードまでの TCP スループット、 B は宛先ノードまでの経路上に存在する各 CH のバッテリー残量の平均を表す．また、 $\omega_1, \omega_2, \omega_3$ により重み付けがされる．本研究において、最もネットワークのスループットに左右されるメトリクスは TCP スループットである．続いてホップ数、バッテリー残量の順に重み付けをしていく．

4 拡張した DSR によるシミュレーション

この章では、本研究で提案したルーティングアルゴリズムのシミュレーションの概要について述べる．

4.1 シミュレーション環境の構築

提案手法の有用性を評価するために使用したツールと実装するプログラムについて説明する．DSR に C++ で新たに記述した、追加モジュールを ns-3 に実装する．

追加モジュール

本研究で DSR に追加または拡張するモジュールについては以下のとおりである．

- 実験用シナリオ
- 優先度を基にした経路選択モジュール
- RTTPL 要求・転送・応答モジュール
- TCP スループット算出モジュール

²Round Trip Time and Packet Loss ratio

シミュレーション環境

シミュレーションで使用するパラメータを表3に示す。

表3 実験シナリオの設定パラメータ

シミュレート時間	100s
ノード数	25,50,75,100
ノード速度	0~10m/s
通信エリア	1000*1000m
各ノードの帯域幅	1Mbps
通信プロトコル	TCP
移動モデル	Random Way Point
MACプロトコル	IEEE802.11b

シミュレーションでは、ノードを分散させるためシミュレーション開始40秒後から通信を開始する。

4.2 シミュレーション結果

表3で示したパラメータ環境で従来のDSRと本研究で提案するDSRをシミュレートする。以下の図2に、時間経過とTCPスループットの変化の関係性を示す。

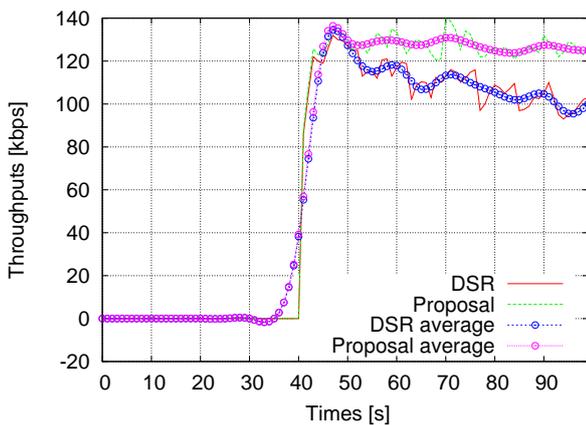


図2 時間経過におけるTCPスループット(ノード数:25)

以下の図3に、ノード数とTCPスループットの変化の関係性を示す。

4.3 シミュレーション結果の考察

図2の結果より、従来のDSRは通信開始から通信終了まで最大28%TCPスループットが減少した。一方、本研究で提案するDSRは、通信開始から通信終了まで最大11%TCPスループットが減少した。また、図2から提案するDSRと従来のDSRを比較して、提案するDSRの方はTCPスループットの乱れが小さいことがわかる。これは、通信が混在したネットワークにおいてもTCPスループットを考慮した経路選択がうまく機能しており、より安定的な経路選択ができていると考えられる。

また、図3の結果より、従来のDSRはノード数の増加に比例してTCPスループットが大きく減少している。特に、ノード数が100台のときのTCPスループットは、従来のDSRが87kbpsに対して提案するDSRは123kbps

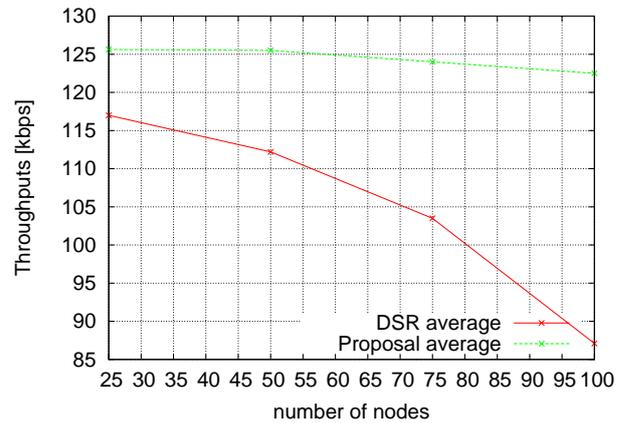


図3 ノード数における平均TCPスループット

に29%向上しており、ノード数が増加した場合においても、安定的なTCPスループットを維持できていることがわかる。

5 おわりに

MANETにおける実験は数多くなされてきた。しかし、どれもUDP通信によるものでTCP通信を考慮した研究は非常に少ない。そのため本研究は、TCP通信のための最適な経路を選択するルーティングアルゴリズムを提案した。ns-3上にTCP通信のための拡張したDSRを実装し、従来のDSRと比較した。結果として、提案したルーティングアルゴリズムは、通信の混在したMANET環境においても安定したTCPスループットを維持することができた。今後の課題としては、実機を用いて実際にMANETを構築し評価をする。IEEE802.11gを用いた場合のノード間距離が45m以上離れると転送率が著しく悪化する問題を解消することで、同じシミュレーションシナリオにおいても安定して、より高いTCPスループットを実現できると考えられる。

参考文献

- [1] Floyd, S. and Fall, K.: Promoting the use of end-to-end congestion control in the Internet, *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, p. 458 (1999).
- [2] Padhye, J., Firoiu, V., Towsley, D. and kurose, J.: Modeling TCP Throughput: A Simple Model and Its Empirical validation, *SIGCOMM Comput. Commun. Rev.*, Vol. 28, No. 4 (1998).
- [3] 浅沼佑紀, 中村嘉隆, 高橋修: MANETにおけるトラフィック状況を考慮したクラスターベースルーティングアルゴリズムの提案, 第76回全国大会講演論文集, Vol. 2014, No. 1, pp. 269-270 (2014).
- [4] 鳴海寛之, 白石陽, 高橋修: MANETにおける高信頼クラスターベースルーティングアルゴリズムの提案と実装・評価, 研究報告システムLSI設計技術, Vol. 2010, No. 15, pp. 1-8 (2010).