

平成25年度 修士論文

邦文題目

通信状態を考慮したアドホックルーティング
プロトコルの提案と評価

英文題目

**A proposal on an Ad-hoc Routing Protocol
considering Traffic Condition and its Evaluation**

情報工学専攻 (学籍番号: 123430038)

三鴨 勇太

提出日: 平成26年01月31日

名城大学大学院理工学研究科

内容要旨

アドホックネットワークでは、端末同士が直接通信し、直接電波の届かない端末へはマルチホップ通信を行う。マルチホップ通信時の経路生成はアドホックルーティングプロトコルによって行われ、標準化されたアドホックルーティングプロトコルやそれを拡張したものが用いられる。しかし、標準化されている多くのプロトコルでは、中継回数が最も少なくなる経路、最短経路を選択しており、最短経路が複数存在する場合に、どの経路を選択するかは規定されていない。そのため、特定の端末に負荷が集中すると、パケットロスが多発し、スループットの低下に繋がるという課題がある。また、標準化されたプロトコルを改造したものを含め、これまで提案されているプロトコルは、IP 網に存在する TCP と UDP という特性の全く異なるふたつの通信タイプに対し、同一の制御を行っており、特性の差を経路探索に生かし切れていないという課題がある。これらの課題に対し、本論文では OLSR (Optimized Link State Protocol) を拡張することにより、TCP と UDP それぞれの特徴を活かすと共に、通信状態を考慮した経路探索を行うアドホックルーティングプロトコル PD-OLSR (Protocol Dependent OLSR) を提案し、シミュレーションによる性能評価を示す。

目次

第1章	はじめに	1
第2章	アドホックネットワーク	3
2.1	概要	3
2.2	既存のアドホックルーティングプロトコルの分類	3
2.3	TCP と UDP に対する挙動の違い	4
第3章	OLSR	6
3.1	隣接ノードの発見	6
3.2	フラッディング方式	6
3.3	トポロジ情報の共有	7
3.4	その他のメッセージ	7
3.5	各ノードが持つ情報	7
3.6	RT の生成	8
3.7	OLSR の課題	8
第4章	PD-OLSR	10
4.1	特徴	10
4.2	トラフィック情報	10
4.3	経路選択	11
第5章	実装	18
5.1	OLSR の改造	18
5.2	情報リポジトリ	19
第6章	評価	21
6.1	シミュレーション条件	21
6.2	TCP	22
6.3	UDP	23
第7章	提案方式の今後の展開	31
7.1	TCP の経路制御	31
7.2	パケットの種類による経路制御切替	31

7.3 リンクメトリックの拡張	32
第8章 まとめ	34
謝辞	35
参考文献	36
研究業績	38

第1章 はじめに

無線 LAN は、配線が不要で端末が自由に移動できるなどの利便性や、スマートフォン、タブレットといった携帯デバイスの登場に伴い、急速に普及してきている。無線 LAN を構築する方法には、端末が AP (Access Point) を介して通信するインフラストラクチャモードと、端末同士で直接通信を行うアドホックモードがある。後者は、災害時やイベント会場など、一時的な無線ネットワーク環境を構築でき、配線が不要であることから、端末の新規参入も容易である。また、AP 間の配線を実線化し、AP によりアドホックネットワークを構築する無線メッシュネットワークにも注目が集まっている [1-5]。

アドホックネットワークの構築には、各端末がアドホックルーティングプロトコルを用いてルーティングテーブル (以下 RT と表記) を生成する必要がある。IETF (Internet Engineering Task Force) により、現在までに多くのアドホックルーティングプロトコルが標準化されているが [6-12]、経路生成の際に中継回数が最小となる経路 (最短経路) を選択することを目的としており、最短経路が複数存在する場合に、どの経路を選択するか規定されていない。そのため、特定のノードにトラフィックが集中すると、パケットロスが多発し、スループットが低下するという課題がある [13]。

この課題に対し、様々な視点から標準化されたプロトコルを拡張したものが提案されている。例えば、ABR (Associativity-Based Routing) [14] では、リンク切断が長時間起こらない、安定した経路を選択する。各ノードは、一定間隔ごとに隣接ノードへビーコンを送信する。より多くのビーコンを受信したノードからなるリンクは、持続性が高いと期待できるため、安定した経路が生成できる。しかし、ノードの移動が少ない環境など、ビーコンの受信回数に差が出ない環境では、スループットの向上が期待できない経路が選択される可能性がある。ETR (Estimated TCP-Throughput Maximization based Routing) [15] では、DSR (Dynamic Source Routing Protocol) [7] を拡張し、宛先への複数の経路候補に対して TCP スループットを予測し、スループットの高い経路を選択する。TCP のスループット予測は、所定のモデル式を用いて計算される。モデル式には、遅延 (Round-Trip Time) および往復パケット損失率 (Round-Trip Packet Loss ratio) の情報が必要であり、これらの情報を収集するために、新たな制御メッセージを設け、一定間隔で送信する。ETR は、TCP スループットのみに着目しており、UDP スループットは考慮していない。また、新たな制御メッセージによるネットワークオーバーヘッドが高くなるという課題がある。

IP ネットワークには、TCP と UDP という、スループット特性が全く異なる通信タイプが存在する。しかし、現在提案されているアドホックルーティングプロトコルでは、両者に対

し、同一の制御を行うことを想定しており、性能を引き出し切れていないという課題がある。

本論文では、アドホックルーティングプロトコルの中でプロアクティブ型の代表的なプロトコルである OLSR (Optimized Link State Routing) [6] を拡張することにより、TCP と UDP で別々に RT を生成し、通信特性を活かした最適な経路選択を可能とするプロトコル PD-OLSR (Protocol Dependent-OLSR) を提案する。TCP と UDP それぞれの RT 生成機能を実装し、シミュレーション評価を行った。その結果、TCP ではドロップパケット数を抑制することができたが、通信中に経路を切り替えることによりスループットが低下することがわかった。UDP では高負荷時のパケット到達率が約 70% から約 100% と劇的に改善できることがわかった。

以下、2 章にアドホックネットワークについて、3 章に OLSR の概要を示す。4 章で PD-OLSR の概要、5 章でシミュレータへの実装について示す。6 章で評価を行い、7 章で提案方式の今後の展開を示す。最後に 8 章でまとめる。

第2章 アドホックネットワーク

2.1 概要

無線 LAN の接続形態にはインフラストラクチャモードとアドホックモードがある。現在広く普及している無線 LAN の形態は前者であり、無線 LAN ルータなどの中継機器を介して端末が通信を行う。一方、アドホックモードでは、中継機器を必要とせず端末同士が直接通信する。アドホックモードにより、端末が自律的に構成するネットワークをアドホックネットワーク (Ad-hoc Network) と呼び、インフラが不要なネットワークとして注目が集まっている。アドホックネットワークには、インフラストラクチャモードにおける中継機器間の接続をアドホックモードで行い、中継機器と端末間をインフラストラクチャモードで接続するメッシュネットワークや、携帯端末など移動するノードを想定した MANET (Mobile Ad-hoc Network)、車々間通信によるネットワーク構築を行う VANET (Vehicle Ad-hoc Network) など様々な形態が存在する。アドホックネットワークでの通信には、特化したルーティングプロトコルであるアドホックルーティングプロトコルを用い、制御メッセージにより情報を収集し RT の生成が必要となる。

2.2 既存のアドホックルーティングプロトコルの分類

アドホックネットワークでは電波到達範囲外のノードと通信するため、各ノードは中継機能を持ち、動的な経路生成を行う必要がある。アドホックネットワークには様々な用途が考えられ、用途に応じたアドホックルーティングプロトコルが存在する。これまでに、様々なアドホックルーティングプロトコルが検討されているが、すべての環境に適するプロトコルは存在しない。これまでに開発されたアドホックルーティングプロトコルは、以下に示すように 3 種類に分類することができる。これらは、その特徴が活かせる環境によって使い分けられる [16–19]。

2.2.1 Proactive 型

Proactive 型のルーティングプロトコルは、通信要求発生前から RT を生成しておく方式であり、通信要求が発生すると即座に通信を開始することができる。各ノードは、ネットワークトポロジ情報を格納するためのテーブルを 1 つ以上持ち、トポロジの変化に応じてネットワーク全体に、更新情報を広告する。ルーティングに必要なテーブル数と、ネットワークト

ポロジの変化を知らせるブロードキャスト方式の違いにより、いくつかのプロトコルが存在する。Proactive 型のルーティングプロトコルの特徴として、通信が発生していない場合でも制御パケットが流れるため、消費電力は大きくなるが、通信を開始する際に遅延が発生しないことから、通信頻度の高いネットワークに適している。例えば、代表的なプロトコルとして OLSR や TBRPF [9] などが挙げられる。

2.2.2 Reactive 型

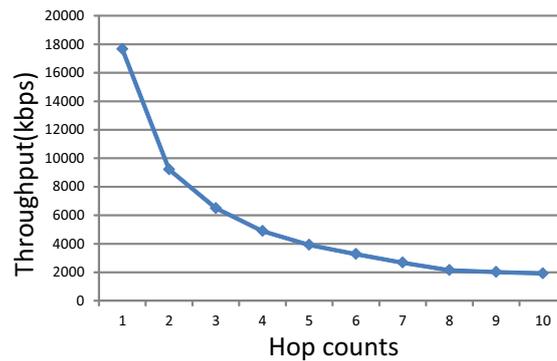
Reactive 型のルーティングプロトコルは、オンデマンド型のプロトコルである。すなわち、あるノードにおいて宛先ノードへの経路が必要になった時点で、ネットワーク内での経路探索プロセスを始動する。このプロセスは、経路が見つかるか、利用可能なすべての経路パターンを探索し終わると終了する。一度経路が発見され、確立すると、宛先へのアクセスが不可能となるか、経路が不要となるまでは、その経路が維持される。Reactive 型のルーティングプロトコルの特徴として、通信時に経路を決定するまでに遅延が発生するが、オンデマンドで経路を構築するため、ノードの移動が頻繁なネットワークに適している。例えば代表的なプロトコルとして AODV [8] や DSR [7] などが挙げられる。

2.2.3 Hybrid 型

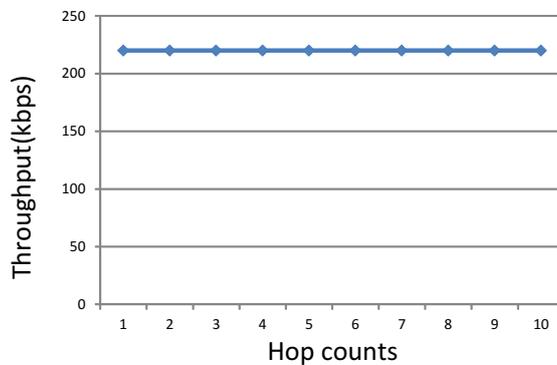
Hybrid 型のルーティングプロトコルは、Proactive 型と Reactive 型の両方の長所を取り入れた複合プロトコルである。ネットワーク内を複数のゾーンに分割し、ゾーン内では Proactive 型のルーティングを行い、定期的な経路情報の更新はゾーン内のノードについてのみ行う。宛先ノードが送信元のゾーン外にある場合は、Reactive 型のルーティングを行い、経路を構築する。Hybrid 型ではこのように両方の特徴を活かすことができるが、ノードが密集するような場合においては、ゾーン内の管理すべきノードが多くなり、トポロジ管理が難しいという課題がある。例えば、代表的なプロトコルとして ZRP [10] などが挙げられる。

2.3 TCP と UDP に対する挙動の違い

IP ネットワークでは、TCP と UDP という特性の全く異なる 2 種類の通信が存在する。TCP では輻輳制御により、順調に ACK が返ってきた場合は、ウィンドウサイズを大きくし、帯域を有効に使おうとする。パケットロスが発生すると、ネットワークに輻輳が発生したと判断し、ウィンドウサイズを小さくする。このようにして、ウィンドウサイズが適切な大きさに調整され、ネットワークのさらなる輻輳を防止する。これに対し、UDP では、端末側が意図した流量のトラフィックがそのままネットワークへ送出される。ネットワーク内でパケットロスが発生したとしても変化はない。



(a) TCP の場合



(b) UDP の場合

図 2.1 マルチホップ通信時のスループット変化

無線ネットワークでは、各端末が同時に電波を送信すると電波同士で干渉が発生する。そのため、RTS/CTS (Request to Send/Clear to Send) といった技術を用いてタイミング制御を行う。マルチホップ通信時には、同一のセッションによる送信であっても、各ホップ間で干渉が発生するため、よりシビアなタイミング制御が必要となる。マルチホップ通信時には、TCP と UDP の通信特性の違いはスループットにも表れる。実際にシミュレーションによって TCP と UDP のマルチホップ通信時のスループットの変化を観測した。図 2.1 に、それぞれ TCP と UDP のホップ数とスループットの関係を示す。ノードを隣接ノードのみと通信可能な距離だけ離して一直線上に並べ、ホップ数を 1~10 の間で変化させた場合のスループットを測定した。TCP ではホップ数の増加と共にスループットが大きく低下する。これは、TCP の輻輳制御により、各ホップでネットワーク帯域を使い切ろうとするためである。これに対し、UDP では、帯域に余裕がある限り、ホップ数増加によるスループット低下は見られない。このことから、UDP では最短経路よりもホップ数を伸ばした経路（冗長経路）でも許容できると考えられる。

第3章 OLSR

プロアクティブ型のプロトコルは，RT を定期的に更新するために送受信される制御メッセージを改造することにより，シンプルに各ノードの情報をネットワーク全体へ通知することが可能であり，その情報をもとに経路上の通信状態を計算することができる．そこで，プロアクティブ型の代表的なプロトコルであり，かつ最も普及している OLSR を提案方式のベースとする．以下に OLSR の原理と RT 生成機能生成方法について説明する．

3.1 隣接ノードの発見

各ノードは，HELLO メッセージを定期的（デフォルト送信間隔 2 秒）に，隣接ノードへブロードキャストする．HELLO メッセージには，自身のアドレス，シーケンス番号，隣接ノードのアドレスなどの情報が記載されている．このため，HELLO メッセージを受信したノードは，隣接ノードのアドレスおよび，隣接ノードの隣接ノード，つまり 2 ホップ先のノード（以降 2 ホップ隣接ノードと表記）のアドレスを得ることができる．また，受信した HELLO メッセージの隣接ノードアドレスの中に自身のアドレスが含まれていれば，自身が送信した HELLO メッセージを隣接ノードが受信したことが確認できる．これは，自身と隣接ノード間で双方向に HELLO メッセージの送受信が可能ということであり，このようなリンクを双方向リンクと呼ぶ．一方，受信した HELLO メッセージの隣接ノードアドレスの中に自身のアドレスが含まれていなければ，そのリンクは非双方向リンクの状態と認識される．これらのリンク状態も，HELLO メッセージに含められブロードキャストされる．

3.2 フラッディング方式

OLSR の最大の特徴は，効率の良いフラッディングの実現である．フラッディングとは，各ノードが自身の情報を，ネットワーク内のすべてのノードへ通知する動作である．3.1 に示すように，通常のフラッディングでは，送信元ノードはメッセージを隣接ノードへブロードキャストする．それを受信した隣接ノードは，ブロードキャストを繰り返し，すべてのノードにメッセージを中継する．同じメッセージを重複して受信した場合は，そのメッセージを破棄する．この方法では，ネットワーク内のノード数が増えると，ブロードキャストによるパケット数が急激に増大し，帯域を圧迫する．OLSR では，必要最低限の中継ノード MPR (Multi-Point Relay) を選択し，MPR として選択された場合のみフラッディング動作を

行うことにより、すべてのノードにメッセージを届ける。各ノードは自身の MPR を選択すると、その情報を HELLO メッセージにより隣接ノードに通知する。これを受信したノードは、自身を MPR として選択しているノードを認識できる。このようなノードを MPR セレクタと呼ぶ。各ノードは、自身の MPR セレクタからのメッセージのみを中継する。このようにして、ブロードキャストの総数を減らし、効率的なフラッディングを実現する。

3.3 トポロジ情報の共有

各ノードは、トポロジ情報を TC (Topology Control) メッセージにより定期的 (デフォルト送信間隔 5 秒) にフラッディングする。TC メッセージを生成するのは MPR のみである。TC メッセージには、自身のアドレス、シーケンス番号、自身の MPR セレクタのアドレスなどの情報が入っている。TC メッセージによってフラッディングされるトポロジ情報は、MPR セレクタから構成されるトポロジのみである。

3.4 その他のメッセージ

OLSR には、HELLO メッセージと TC メッセージの他に、MID (Multiple Interface Declaration) メッセージと HNA (Host and Network Association) メッセージがある。MID メッセージは、ノードが複数のインターフェースを有する場合にのみ使用され、HNA メッセージは、ノードがゲートウェイとして機能する場合に使用される補助的なメッセージである。本論文の提案方式では、MID および HNA メッセージが使用されるような環境は想定しないため、これらの説明は省略する。

3.5 各ノードが持つ情報

各ノードは RT を生成するために、リンク集合、隣接ノード集合、2 ホップ隣接ノード集合、MPR 集合、MPR セレクタ集合、トポロジ集合および複製集合の 7 つテーブルから成るリポジトリを持つ。これらのテーブルは HELLO メッセージおよび TC メッセージによって生成、更新される。

リンク集合は、ローカルノード自身のアドレス、隣接ノードのアドレス、リンクが双方向とみなされる時間、レコードの生存時間から構成される。隣接ノード集合は、隣接ノードのアドレス、リンクが双方向か非双方向であるかの状態、MPR として選択されるための指標 (Willingness) から構成される。2 ホップ隣接ノード集合は、隣接ノードのアドレスと双方向の 2 ホップ隣接ノードのアドレス、レコードの生存時間から構成される。MPR 集合は、MPR として選択されたノードのアドレスとレコードの生存時間から構成される。MPR セレクタ集合は、MPR セレクタとして選択されたノードのアドレスとレコードの生存時間から

構成される。トポロジ集合は、宛先となるノードのアドレス、宛先へ1ホップで到達できるノードのアドレス、レコードの生存時間から構成される。複製集合は、受信したメッセージの重複した処理を避けるために設けられるテーブルである。

3.6 RTの生成

OLSRのRTは、宛先ノード (Dest)、Destへの次ホップノード (Next)、Destまでの距離 (Dist) から構成され、各 Dest に対して1つの経路を保持する。図3.2にOLSRの経路生成手順を示す。図3.2(a)のように、簡単のため19台のノードが規則的に配置されており、電波到達範囲は隣接ノードまでとしている。図3.2(b)において、2つのRTはノードaが持つRTであり、左側はノードbからsのうち、ノードbからqまでの経路が途中まで生成された状態、右側はさらにノードrとsまでの経路が生成し終わった状態を示す。以下に、左側のRTから右側のRTが生成される過程を示す。

左側のRTにDestがノードrとなる経路が新たに追加されるとき、ノードrへ到達可能なノードrの隣接ノードn, o, q, sを既に生成されているRTの上から探索したとき最初に見つかるノードnに注目し、ノードnのNext: dをノードrのNextとして登録する。さらにノードnのDist: 3に1を足した4をノードrのDistとして登録する。Destがsの経路も同様に追加される。上記の方法で、すべてのノードのRTが生成されると、ノードrへの経路が決まり、図3.2(a)内に示す経路 [a d i n r] という最短経路が完成する。このようにOLSRでは、単純に最初に発見された最短経路が選ばれる。

3.7 OLSRの課題

もし、図3.2の状況において、ノードiからhへの通信が既に行われている状態で、ノードbからrへの通信が生成された経路を用いて行われると、パケットロスが発生しスループットが低下する可能性がある。このようにOLSRでは、新たなトラフィックが発生したときに、効率の良い経路選択ができないという課題がある。

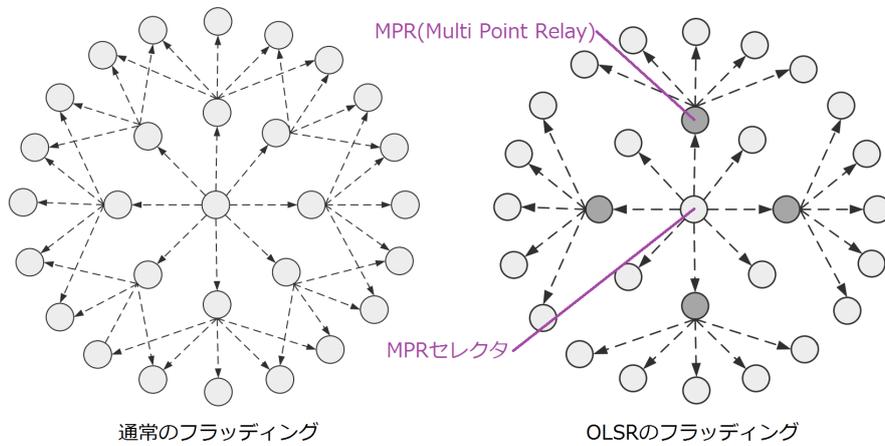
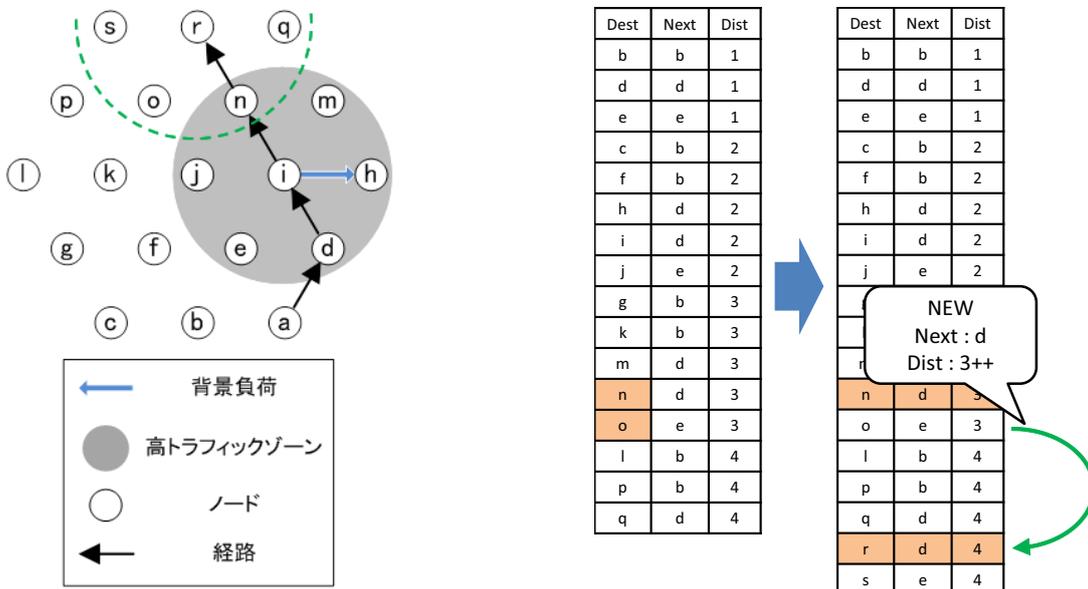


図 3.1 OLSR のフラッディング



(a) ネットワーク例

(b) ノード a の RT の変化

図 3.2 OLSR の RT 生成

第4章 PD-OLSR

本論文では、OLSR をベースにした、通信状態を考慮してプロトコルごとに効率の良い経路選択を行うアドホックルーティングプロトコル PD-OLSR (Protocol Dependent OLSR) を提案する。本提案では、OLSR の基本部分はそのままだけに、経路選択指標については、TCP と UDP の特性に応じてそれぞれ決定する。

4.1 特徴

PD-OLSR は、次のような特徴を持つアドホックルーティングプロトコルである。

(1) TCP と UDP で別々の RT を生成

異なる特性を持つ TCP と UDP に対し、別々の RT を用いて通信制御を行うことにより、通信特性を活かした経路選択を行う。

(2) 通信状態による経路選択

ノードの通信状態を計測し、ノードへのトラフィック集中を抑制する。各ノードは、トラフィック情報を常に計測する。計測した情報は制御メッセージに載せ広告、共有することで経路選択に用いる。このとき、制御メッセージの送受信、ノードが保持する情報の更新といった基本部分は OLSR のものをそのまま利用し、制御メッセージにトラフィック情報を追加するものとする。

4.2 トラフィック情報

TCP 用と UDP 用の別々の RT のために、経路選択指標として用いるトラフィック情報も分けて検出する。TCP 用を TCP セッション数、UDP 用を UDP トラフィックと定義する。TCP セッション数は、自身が検出する 1 秒ごとの TCP のセッション数の合計とし、UDP トラフィックは、自身が検出する 1 秒ごとの UDP のキャリアの総量とする。さらにこれらには、自身が送信元となる通信についても含めるものとする。

4.3 経路選択

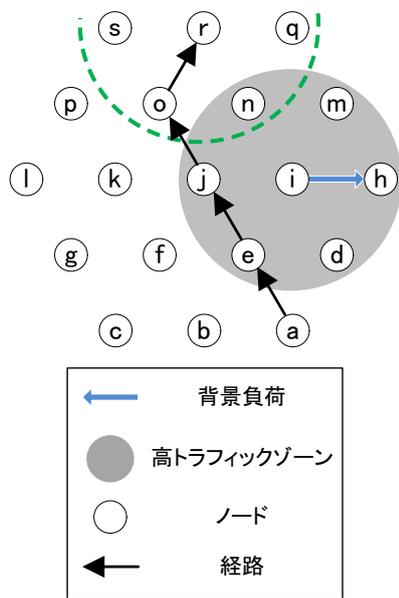
PD-OLSR では、TCP 用と UDP 用の RT を別々に生成する。以下に経路探索に用いるアルゴリズムおよび RT 生成について説明する。

4.3.1 TCP 用 RT 生成

TCP では、最短経路の中から最適な経路を選択する。図 4.1 に、RT 生成の様子を示す。ネットワーク例は図 3.2(a) と同様とし、ノード i から h への背景負荷が TCP の場合を考える。このとき、各ノードの TCP セッション数は図 4.1(b) のようになる。また、PD-OLSR の RT の構成は、OLSR と同様に Dest, Next, Dist であるが、説明のため、自ノードから Dest までの経路コスト (Cost) を併せて記載している。

図 4.1(c) において、2 つの RT はノード a が持つ RT であり、左側はノード b から s のうち、ノード b から q までの経路が途中まで生成された状態、右側はさらにノード r と s までの経路が生成し終わった状態を示す。以下に、左側の RT から右側の RT が生成される過程を示す。

左側の RT に Dest がノード r となる経路が新たに追加される場合では、ノード r へ到達可能なノード r の隣接ノード n, o, q, s を既に生成されている RT の上から探索し、Dist が最も小さく、その中で Cost が最も小さいノード o が選択される。ノード o の Next: e をノード r の Next として、ノード o の Dist: 3 に 1 を足した 4 をノード r の Dist として RT に登録する。また、ノード r の Cost は、ノード o の Cost: 2 にノード o が検出している TCP セッション数: 0 を足した 2 となる。上記の方法で、すべてのノードの RT が生成されると、ノード r への経路が決まり、図 4.1(a) 内に示す経路 $[a \ e \ j \ o \ r]$ という最短経路が完成する。このように PD-OLSR の TCP 用の RT 生成では、最短経路の中から最もコストの小さい経路が選ばれる。



Node	TCP Session
a	0
b	0
c	0
d	1
e	1
f	0
g	0
h	1
i	1
j	1
k	0
l	0
m	1
n	1
o	0
p	0
q	0
r	0
s	0

(a) ネットワーク例

(b) 各ノードの通信状態

Dest	Next	Dist	Cost
b	b	1	0
d	d	1	1
e	e	1	1
c	b	2	0
f	b	2	0
h	d	2	2
i	d	2	2
j	e	2	2
g	b	3	0
k	b	3	0
m	d	3	3
n	d	3	3
o	e	3	2
l	b	4	0
p	b	4	0
q	d	4	3



Dest	Next	Dist	Cost
b	b	1	0
d	d	1	1
e	e	1	1
c	b	2	0
f	b	2	0
h	d	2	2
i	d	2	2
j	e	2	2
g	b	3	0
k	b	3	0
m	d	3	3
n	d	3	3
o	e	3	2
l	b	4	0
p	b	4	0
q	d	4	3
r	e	4	2
s	e	4	2

NEW
 Next : e
 Dist : 3++
 Cost : 2+0



(c) ノード a の RT の変化

図 4.1 TCP 用 RT 生成

4.3.2 ダイクストラ法

PD-OLSRのUDP用RT生成では、アルゴリズムとしてダイクストラ法 [20] を用いる。ダイクストラ法は、グラフ理論における最短経路探索アルゴリズムのひとつである。ノード間のリンクメトリックをもとに、最小コストとなる経路を得ることができる。以下に、ダイクストラ法の動作を説明する。

(i) 経路探索

図 4.2 を用いて、ダイクストラ法の経路探索の様子を示す。各ノードについて、そのノードの状態（未探索/探索済み/確定）、自ノードからのコストおよび直前に経由したノードの情報を記録する。コストは、直前ノードのコストに、直前ノードからのリンクのメトリックを加算するものとする。

Step:0 自ノードのコストを 0 とする。

Step:1 自ノードから直接到達可能な隣接ノードを探索する。隣接ノードを探索済みノードとする。

Step:2 探索済みノードの中から最もコストの小さいノードを確定ノードとする。

Step:3 Step:2 で確定したノードから到達可能なノードを探索する。

Step:4 新たに探索したノードが、未探索ノードの場合は、探索済みノードとし、コストおよび経路を登録する。探索済みノードの場合は、既に記録されているコストと探索時のコストを比較し、探索時のコストの方が小さければ、コストおよび経路を更新する。

Step:5 Step:2 から Step:4 を繰り返す。すべてのノードが確定した場合、終了する

(ii) リンクメトリック

リンクメトリックは、経路コストの計算に用いる各リンクのコストである。PD-OLSRでは、ノードのトラフィック情報をもとに経路を選択する。しかし、リンクメトリックをトラフィック情報のみで求める場合、過剰に迂回する経路を選択する可能性がある。そこで、ホップ数に対してもコストを設定することにより、その大きさによって迂回度を決定できるようにする。リンクメトリックは、リンクの送信元ノードのトラフィック情報とホップ数コストの和によって求める。ホップ数コストを大きくすることにより経路の迂回度は小さくなり、逆にホップ数コストを小さくすることにより迂回度は大きくなる。ホップ数コストを十分大きくすると、最短経路の中から最適な経路選択をする方式と等価となる。

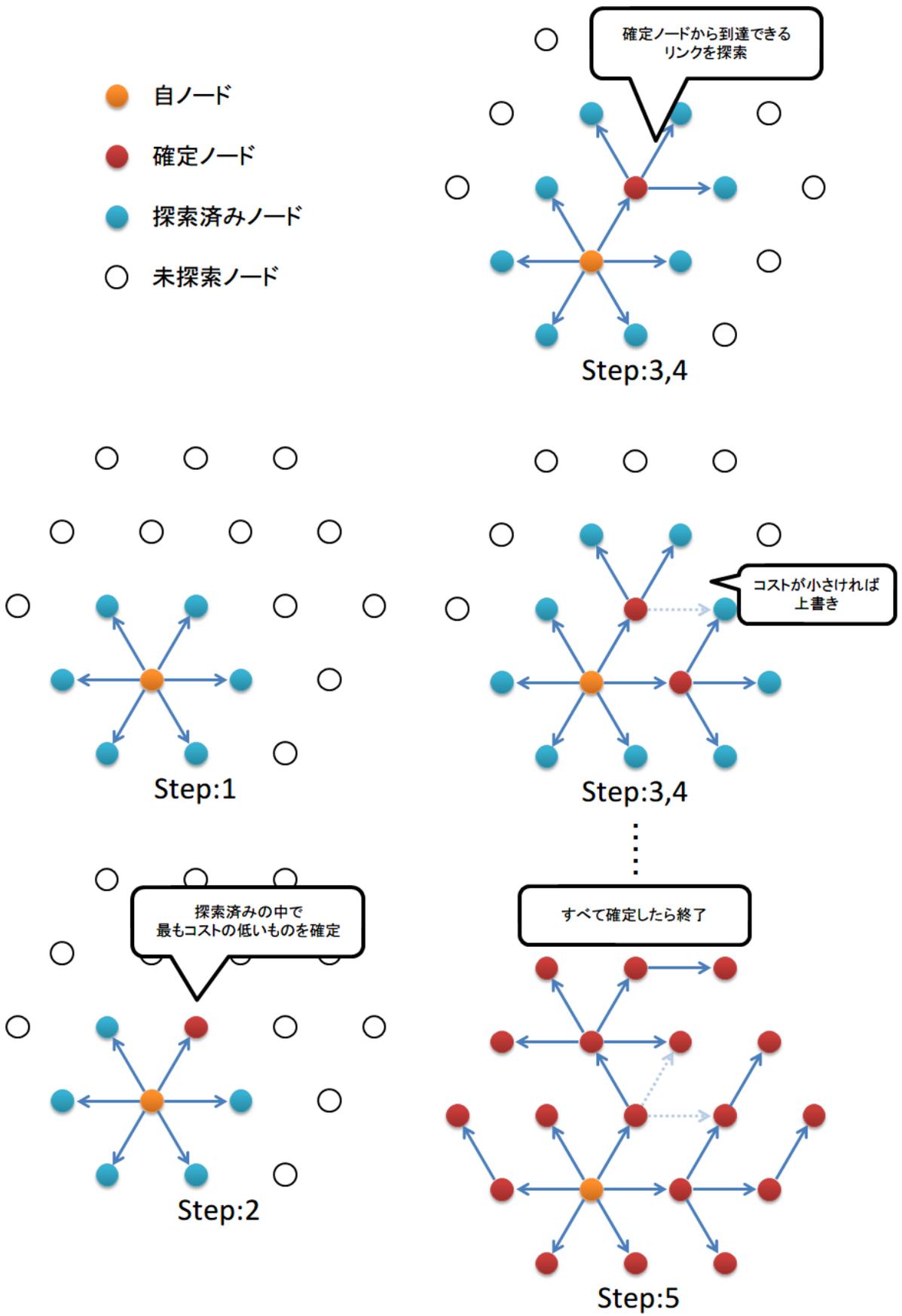


図 4.2 ダイクストラ法

4.3.3 UDP 用 RT 生成

UDP では、冗長経路も許容できることから、取り得るすべての経路の中からダイクストラ法を用いて最適な経路を選択する。以下に、RT 生成の様子を示す。ネットワーク例：図 4.3(a) は図 3.2(a) と同様とし、ノード i から h への背景負荷が UDP の場合を考える。このとき、背景負荷による UDP トラフィックを 4 とすると、各ノードの UDP トラフィックは図 4.3(b) のようになる。

図 4.4 にノード a が RT を生成する様子を示す。左側の図は、ノード a から探索される経路の様子を示し、右側の 2 つのテーブルは経路探索に用いる探索済みノードリスト (Search List) および RT である。Search List は、Dest, Cost, Dist, 直前ノード (From), 確定ノードフラグ (Flg) から構成される。RT の構成要素は、OLSR のものと同様である。

- (1) 自ノード： a の隣接ノードである b, d, e が探索される。このとき経由するノードは自ノードのみとであることから、Cost は 0 となる。隣接ノードを確定ノードとし、RT に Dest, Next, Dist を追加する。隣接ノードから到達可能な 2 ホップ隣接ノードを探索する。このとき、既に探索済みのノードに対しては、Cost がより小さいときのみ更新を行い、それ以外は不採用経路となる。
- (2) 未確定の探索済みノードの中から最初に見つかる最も小さいコストのノード c を確定し、RT に追加する。ノード c から到達できるノードを探索し、Search List に追加する。
- (3) 同様にノード f を確定、RT に追加し、ノード f から到達できるノードを探索する。
- (4) この動作をすべてのノードが確定するまで繰り返す。

以上の動作により、すべてのノードの RT が生成されると、ノード r への経路が決まり、図 4.3(a) 内に示す経路 $[a \ b \ f \ k \ o \ r]$ という経路が完成する。このように、PD-OLSR の UDP 用の RT 生成では、すべての経路の中から最もコストの小さい経路が選ばれる。

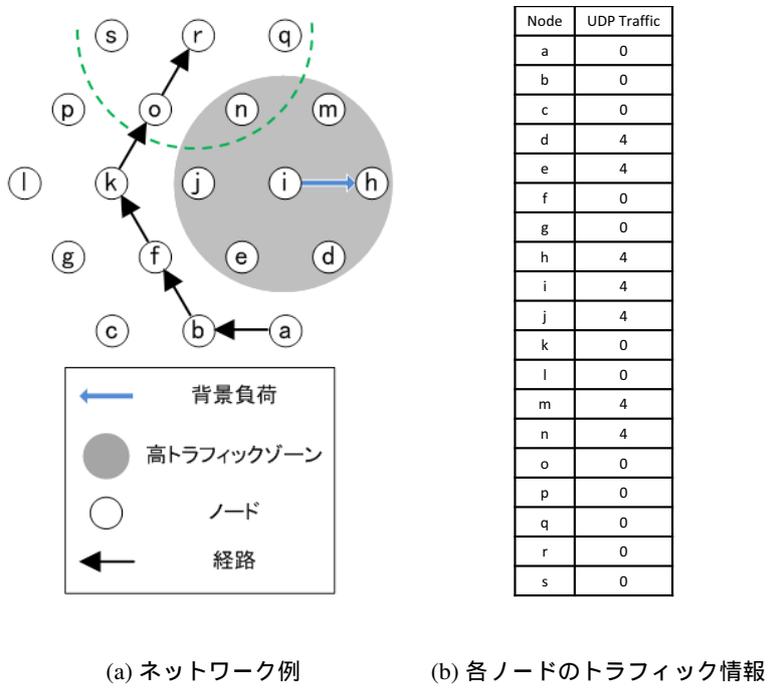
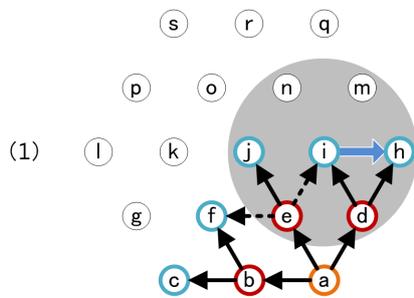
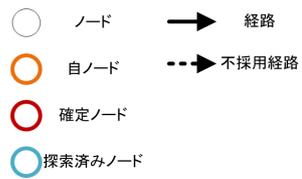
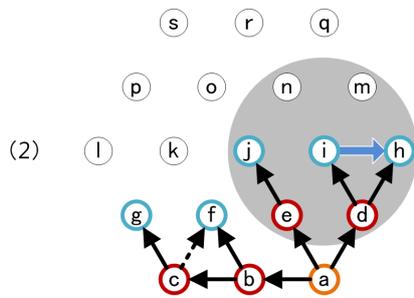


図 4.3 UDP 用 RT 生成時のネットワーク例



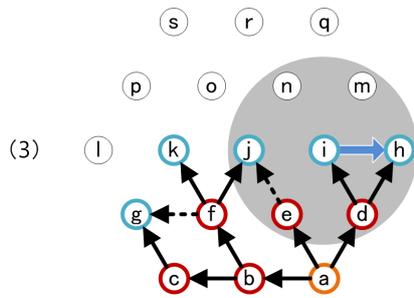
Dest	Cost	Dist	From	Flg
b	0	1	a	1
d	0	1	a	1
e	0	1	a	1
c	0	2	b	0
f	0	2	b	0
h	4	2	d	0
i	4	2	d	0
j	4	2	e	0

Dest	Next	Dist
b	b	1
d	d	1
e	e	1



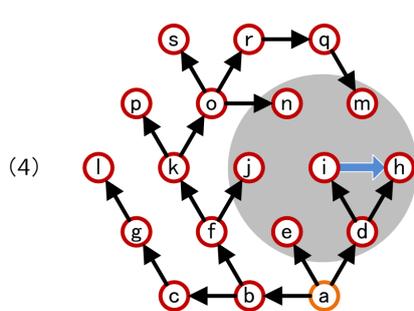
Dest	Cost	Dist	From	Flg
b	0	1	a	1
d	0	1	a	1
e	0	1	a	1
c	0	2	b	1
f	0	2	b	0
h	4	2	d	0
i	4	2	d	0
j	4	2	e	0
g	0	3	c	0

Dest	Next	Dist
b	b	1
d	d	1
e	e	1
c	b	2



Dest	Cost	Dist	From	Flg
b	0	1	a	1
d	0	1	a	1
e	0	1	a	1
c	0	2	b	1
f	0	2	b	1
h	4	2	d	0
i	4	2	d	0
j	0	3	f	0
g	0	3	c	0
k	0	3	f	0

Dest	Next	Dist
b	b	1
d	d	1
e	e	1
c	b	2
f	e	2



Dest	Cost	Dist	From	Flg
b	0	1	a	1
d	0	1	a	1
e	0	1	a	1
c	0	2	b	1
f	0	2	b	1
h	4	2	d	1
i	4	2	d	1
j	0	3	f	1
g	0	3	c	1
k	0	3	f	1
l	0	4	g	1
o	0	4	k	1
p	0	4	k	1
n	0	5	o	1
r	0	5	o	1
s	0	5	o	1
q	0	6	r	1
m	0	7	q	1

Dest	Next	Dist
b	b	1
d	d	1
e	e	1
c	b	2
f	e	2
h	d	2
i	d	2
j	f	3
g	b	3
k	b	3
l	b	4
o	b	4
p	b	4
n	b	5
r	b	5
s	b	5
q	b	6
m	b	7

図 4.4 UDP 用 RT 生成

第5章 実装

PD-OLSRのRT生成機能をネットワークシミュレータ ns-2 [21] に実装した。ベースとなるOLSRは ns-2 には標準では含まれないため、ns-2 向けのオープンソースである UM-OLSR [22] を用いた。本章では、実装内容について示す。

5.1 OLSRの改造

UDP用RT生成を例にしたPD-OLSRのフローを図5.1に示す。OLSRでは、主にHELLOとTCというふたつの制御メッセージによって情報を収集し、RT生成を行っており、PD-OLSRにおいても同様の手順を踏む。以下に、それぞれ処理における改造内容を示す。図中の番号は以下の番号に対応している。

(1) 制御メッセージの送信

- HELLOメッセージとTCメッセージに送信元ノード自身のUDPトラフィックを追加

(2) リンク集合の更新

- HELLOメッセージの送信元ノードと一致する隣接ノードのレコードに送信元ノードのトラフィック情報を記録
- 一致するレコードが存在しないときは、新たに送信元ノードを隣接ノードとするレコードを生成

(3) 隣接ノード集合と2ホップ隣接ノード集合の更新

- (2)の更新と対応する隣接ノードのレコードにトラフィック情報を記録

(4) トポロジ集合の更新

- TCメッセージの送信元ノードと一致する宛先ノードのレコードにトラフィック情報を記録
- 一致する宛先ノードが存在しないときは、新たに送信元ノードを宛先ノードとするレコードを生成

(5) RT生成

- 4.3節に示す方法でRTを生成

5.2 情報リポジトリ

図 5.2 に制御メッセージと情報リポジトリの関係を、図 5.3 に改造した各リポジトリが保持する情報を示す。HELLO メッセージを受信したノードは、リンク集合、2 ホップ隣接ノード集合、MPR セレクタ集合、複製集合を更新する。また、リンク集合、2 ホップ隣接ノード集合の更新に伴い、隣接ノード集合と MPR 集合も更新する。一方、TC メッセージを受信したノードは、トポロジ集合と複製集合を更新する。これらの更新されたテーブルを元に、新しい HELLO メッセージ、及び TC メッセージを生成する。MPR (Multi Point Relay) 集合とは、OLSR の特徴のひとつであるフラッディングを効率的に行うために管理するノードの集合、MPR セレクタ集合とは、自身が MPR 集合に含まれる場合に自身を MPR として選択しているノードの集合である。MPR は、隣接ノード集合に含まれる Willingness をもとに決定する。今回は、リンク集合、隣接ノード集合、2 ホップ隣接ノード集合およびトポロジ集合に赤字で示すノードのトラフィック情報を追加した。なお、追加した情報を含めた情報リポジトリは、制御メッセージに載せて広告される。

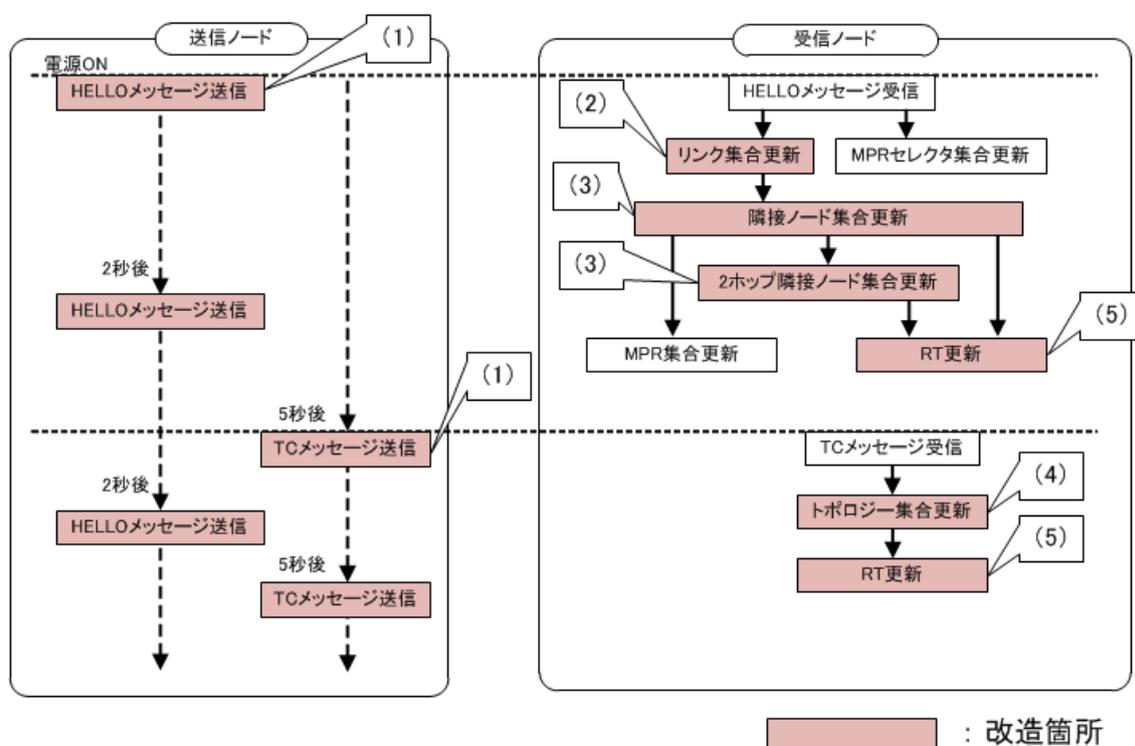


図 5.1 OLSR の改造箇所

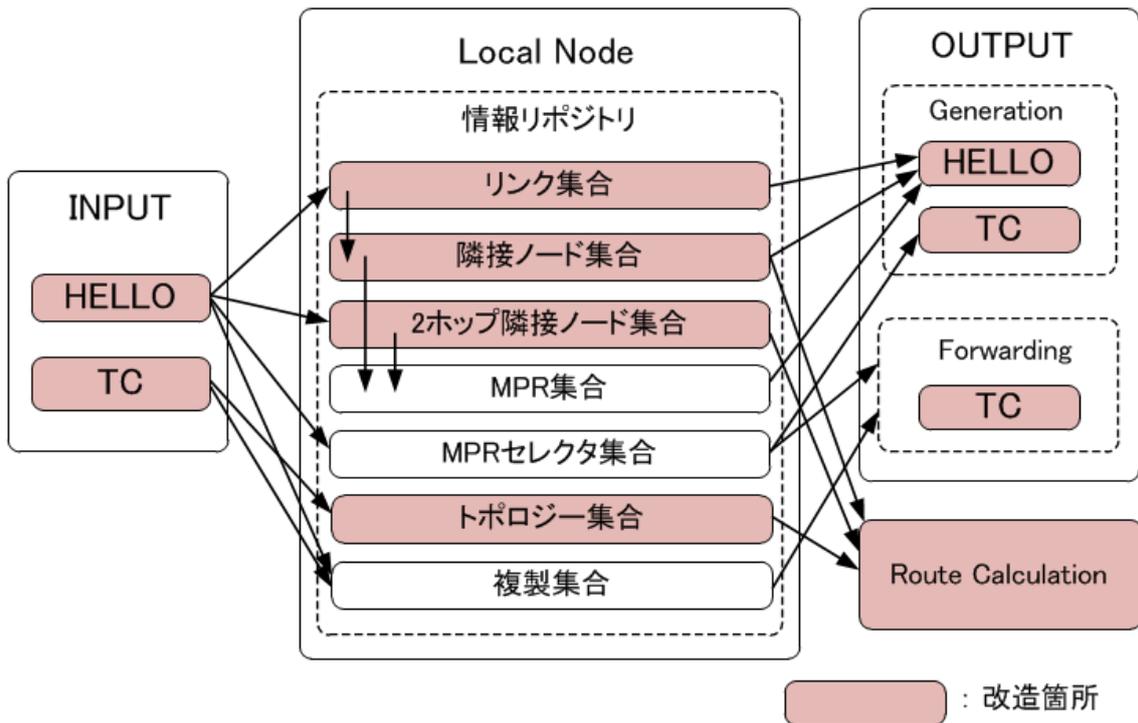


図 5.2 制御メッセージと情報リポジトリの関係

<p style="text-align: center;">リンク集合</p> <p>自ノードインターフェース 隣接ノード リンクタイプ 有効期間 隣接ノードトラフィック</p>	<p style="text-align: center;">2ホップ隣接ノード集合</p> <p>隣接ノード 2ホップ隣接ノード 有効期間 隣接ノードトラフィック</p>
<p style="text-align: center;">隣接ノード集合</p> <p>隣接ノード リンクタイプ 隣接ノードタイプ Willingness 有効期間 隣接ノードトラフィック</p>	<p style="text-align: center;">トポロジ集合</p> <p>宛先ノード 到達可能ノード 到達可能ノードトラフィック 有効期間 シーケンス番号</p>

太字: 追加情報

図 5.3 情報リポジトリ

第6章 評価

5章で述べた PD-OLSR における UDP 用および TCP 用の RT 生成機能についてのシミュレーション評価を行った。

6.1 シミュレーション条件

表 6.1 にシミュレーション条件を示す。ノード数は 37 台，図 6.1 のように等間隔に配置されており，電波到達範囲は隣接ノードまでとする。シミュレーション開始 1 分後から 1 分ごとに 10 本ずつ負荷となるセッションを増加させ計 5 分間，セッションを 40 本まで増加させた。負荷が TCP の場合と UDP の場合，それぞれについて OLSR と PD-OLSR を用いて各 10 回ずつ行った。

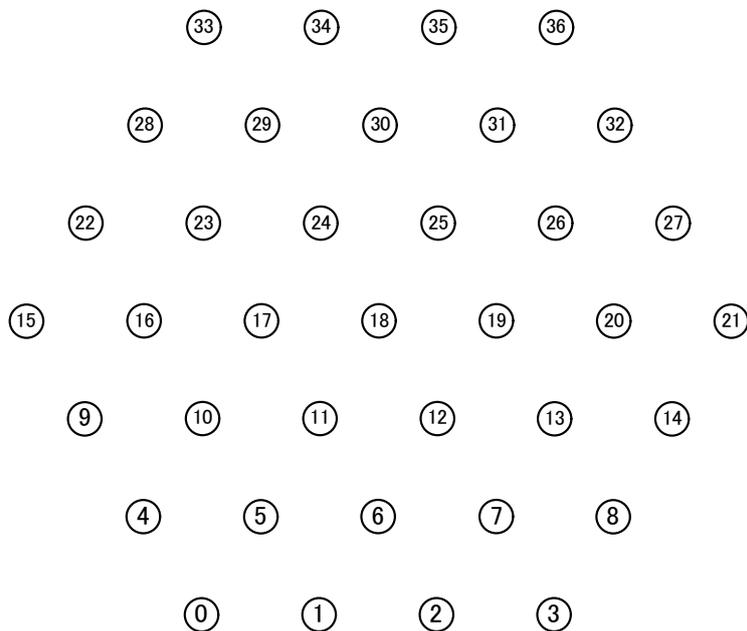


図 6.1 ノード配置

表 6.1 シミュレーション条件

ネットワーク	
形態	アドホックネットワーク
通信規格	IEEE802.11g
ノード数 [台]	37
電波到達範囲	隣接ノード
通信組	2台1組, ランダム
セッション数 [本]	10~40
通信	
UDP	
通信タイプ	CBR
パケットサイズ [Byte]	200
レート [kbps]	64
TCP	
通信タイプ	FTP
パケットサイズ [Byte]	1000
ウィンドウサイズ	15~1023

6.2 TCP

6.2.1 リンクメトリックの設定

PD-OLSRにおけるリンクメトリックは、リンクの送信元方向のノードのTCPセッション数とした。

6.2.2 評価結果

TCPの合計スループットを図6.2と表6.2に、ドロップパケット数を図6.3と表6.3に、パケット到達率を図6.4と表6.4に示す。セッション数10~40本のすべての環境において、特に低負荷時にOLSRと比較してPD-OLSRのスループットが低くなった。しかし、PD-OLSRは、OLSRと比較してドロップパケット数で見ると低くなっていることや、パケット到達率が高いことから、スループット低下の主な原因は、パケットのドロップがではなく送信されるパケット量が少ないことだと考えられる。PD-OLSRでは、トラフィック情報が更新されると、通信中であっても経路を動的に切り替える。通信中に経路を切り替えると、パケットの追い越しが発生し、輻輳制御の働きによってスループットが低下する。このことが、今回の評価におけるスループット低下の原因と考えられる。

6.3 UDP

6.3.1 リンクメトリックの設定

リンクメトリックは式 6.1 のように求める．ここで， L をリンクメトリック， T_{src} をリンク送信元ノードのトラフィック， H をホップ数コストとする．

$$L = T_{src} + H \quad (6.1)$$

また，今回の評価におけるホップ数コスト H は， α を迂回度係数， T_{max} をネットワーク内のトラフィック最大値とすると，式 6.2 のように求める．経路の迂回度に関わるホップ数コストは， T_{max} に依存する．また，経路の迂回度は α によって調整する．このとき， T_{max} は各ノードの持つトポロジ情報内のトラフィック情報から最大値を求めるものとする．

$$H = \alpha T_{max} \quad (6.2)$$

つまり，リンクメトリックは，式 6.3 のように求める．

$$L = T_{src} + \alpha T_{max} \quad (6.3)$$

6.3.2 評価結果

UDP のパケット到達率を図 6.6 と表 6.5 に，ドロップパケット数を図 6.7 と表 6.6 に示す．PD-OLSR の場合では，迂回度係数 α を 0.5～6 の範囲を評価した．係数が大きくなるほど迂回度は小さくなる． α による平均ホップ数の変化を図 6.5 に示す．

OLSR では，UDP の本数が多くなるほどパケット到達率が低下し，40 本するときパケット到達率が 70%弱となった．PD-OLSR ではどの場合でもほぼ 100%と劇的に改善することができた．しかし，PD-OLSR において α を 0.5 とした場合に，UDP が 10 本である場合にも到達率が低下した．このことから，過剰な迂回は低負荷時にパケット到達率を低下させることがわかった．また， α が 1～6 の範囲でのパケット到達率の差がほぼ見られなかったことから，ドロップパケット数について比較してみると，若干の差が見られ α が 1 の場合が最も少なくなり，UDP が 40 本の場合では OLSR と比較して約 74%の改善率となった．さらに，UDP のアプリケーション，例えば IP 電話に用いられる VoIP などでは，通信のリアルタイム性が重視される．そこで，送信元宛先間の平均遅延を測定すると，図 6.8 と表 6.7 のようになった．PD-OLSR では，どの場合も平均遅延が OLSR より 1ms 以上短くなった．このことから，冗長経路であっても負荷の小さい経路を選択することで，OLSR のような単純な最短経路よりも遅延を小さくできることがわかった．しかし， α の大きさが遅延時間と比例していないことから，両者の関係については更なる検証が必要である．

今回測定した α の範囲では、 α を 1 の場合が最も高い性能となったが、この値が必ずしも最適な値とは限らない。経路をどの程度迂回させたときに性能が最大となるかは、ネットワークポロジに依存するものと考えられる。パケット到達率、ドロップパケット数および遅延の観点から、例えばネットワーク内のノード数や隣接ノード数、ノードの移動速度などを用いて α を決定する方法について検討を行う必要がある。また、 α はノードごとに設定できることから自身の通信状態や位置・移動状態をもとに算出することも可能である。

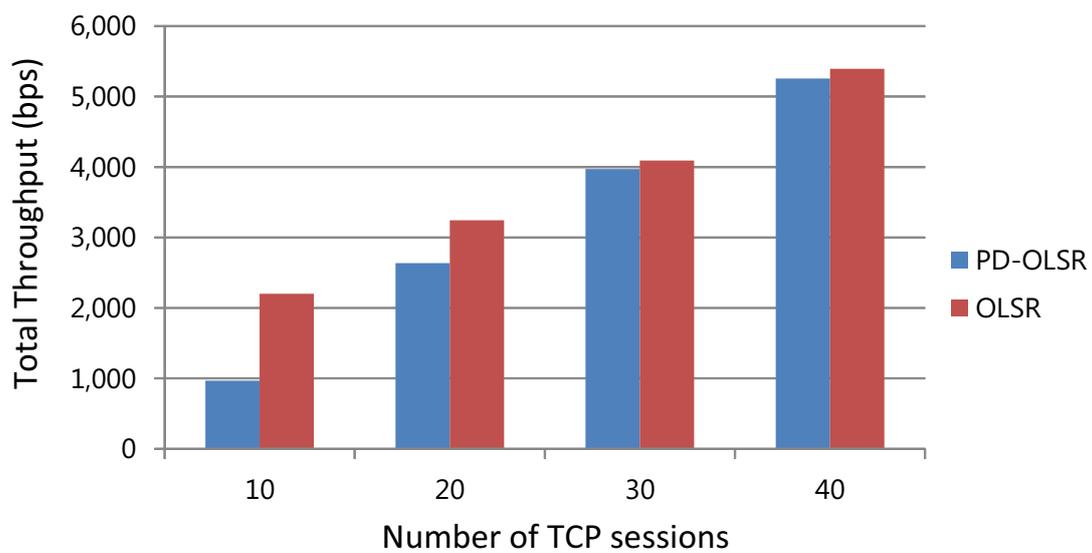


図 6.2 TCP スループット比較

表 6.2 TCP スループット比較

	Number of TCP sessions			
	10	20	30	40
PD-OLSR	967	2,638	3,971	5,257
OLSR	2,200	3,245	4,094	5,394

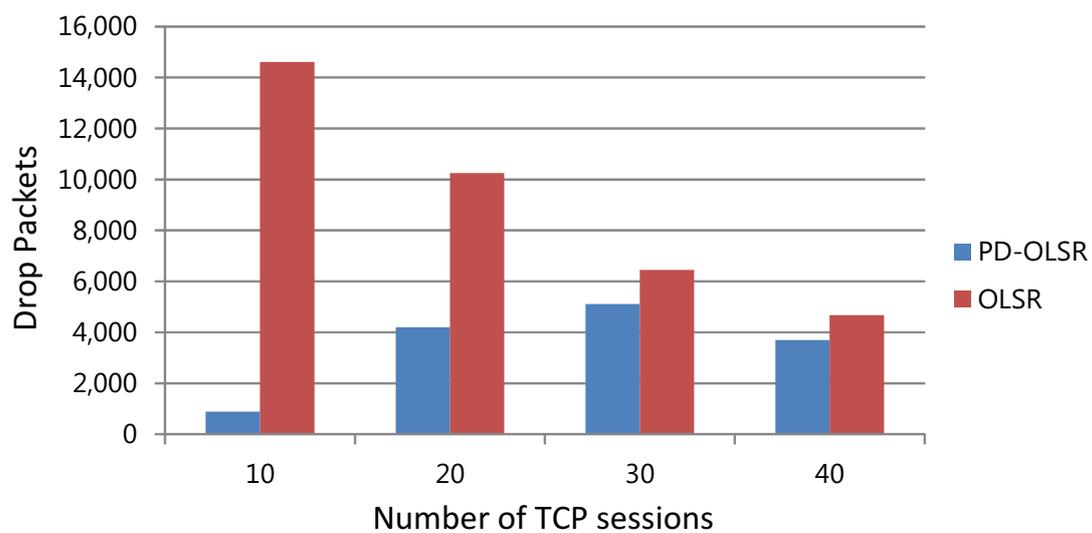


図 6.3 TCP ドロップパケット比較

表 6.3 TCP ドロップパケット比較

	Number of TCP sessions			
	10	20	30	40
PD-OLSR	967	2,638	3,971	5,257
OLSR	2,200	3,245	4,094	5,394

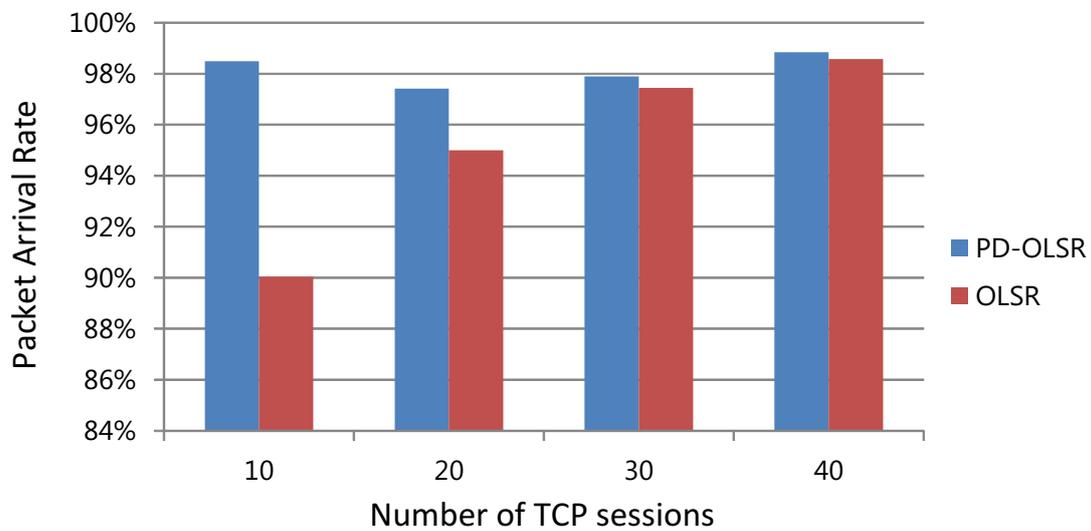


図 6.4 TCP パケット到達率比較

表 6.4 TCP パケット到達率比較

	Number of TCP sessions			
	10	20	30	40
PD-OLSR	98.49%	97.42%	97.90%	98.84%
OLSR	90.03%	95.00%	97.44%	98.58%

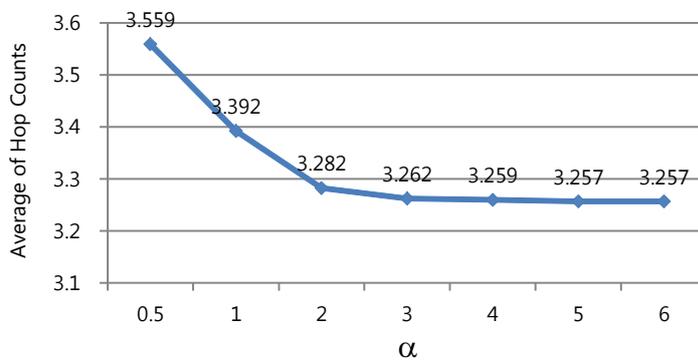


図 6.5 α によるホップ数の変化

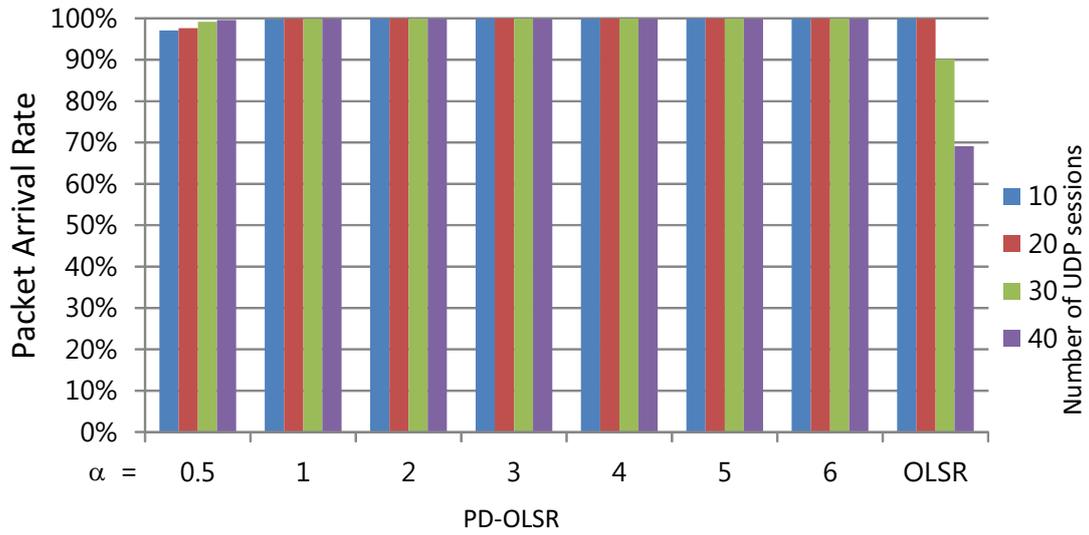


図 6.6 UDP パケット到達率比較

表 6.5 UDP パケット到達率比較

		Number of UDP sessions			
		10	20	30	40
PD-OLSR	0.5	97.07 %	97.65 %	99.15 %	99.63 %
	1	99.81 %	99.89 %	99.93 %	99.94 %
	2	99.96 %	99.98 %	99.96 %	99.94 %
	3	99.95 %	99.98 %	99.96 %	99.93 %
	4	99.96 %	99.98 %	99.97 %	99.93 %
	5	99.97 %	99.98 %	99.96 %	99.93 %
	6	99.97 %	99.98 %	99.96 %	99.94 %
OLSR	-	99.96 %	99.94 %	90.10 %	69.12 %

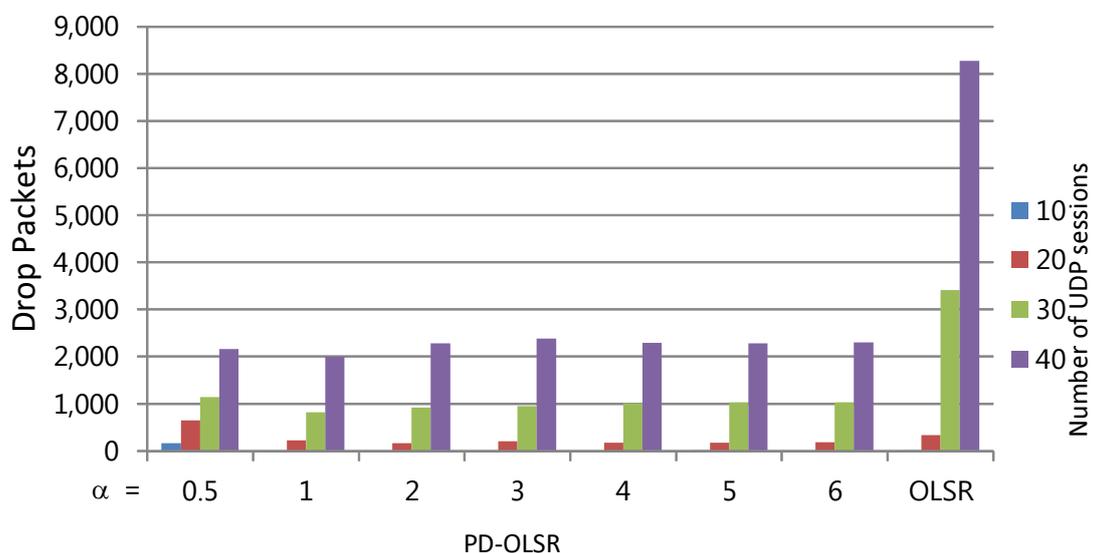


図 6.7 UDP ドロップパケット比較

表 6.6 UDP ドロップパケット数比較

		Number of UDP sessions			
		10	20	30	40
PD-OLSR	0.5	163 (-1153.85 %)	646 (-93.99 %)	1,144 (66.46 %)	2,164 (73.86 %)
	1	10 (23.08 %)	228 (31.53 %)	823 (75.87 %)	2,002 (75.82 %)
	2	6 (53.85 %)	170 (48.95 %)	924 (72.91 %)	2,282 (72.43 %)
	3	10 (23.08 %)	202 (39.34 %)	953 (72.06 %)	2,386 (71.18 %)
	4	6 (53.85 %)	178 (46.55 %)	1,016 (70.21 %)	2,292 (72.31 %)
	5	15 (-15.38 %)	178 (46.55 %)	1,039 (69.54 %)	2,287 (72.37 %)
	6	6 (53.85 %)	181 (45.65 %)	1,031 (69.77 %)	2,308 (72.12 %)
OLSR	-	13	333	3,411	8,278

() 内は改善率

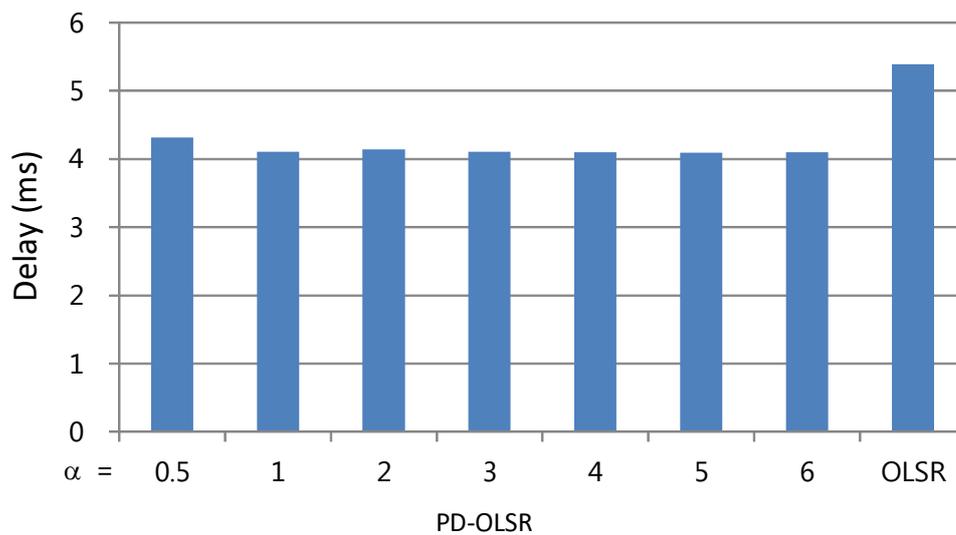


図 6.8 UDP の送信元宛先間平均遅延

表 6.7 UDP の送信元宛先間平均遅延

	Delay(ms)
PD-OLSR 0.5	4.316
1	4.109
2	4.143
3	4.107
4	4.103
5	4.093
6	4.098
OLSR -	5.393

第7章 提案方式の今後の展開

7.1 TCPの経路制御

TCPのRTでは、UDPのRTと同じように最新のトラフィック情報をもとに更新を行っている。この場合、通信中に経路が切り替わることになり、パケットの追い越しが発生する可能性がある。TCPにおける輻輳制御では、パケット追い越しはパケットロスとして認識され、ウィンドウサイズが縮小される。つまり、通信中の経路切替によって送信量、スループットが低下する可能性がある。そのため、TCPでは通信中は同じ経路を使い続ける必要がある。

通信中に経路を固定する機能は、これまでの経路制御を拡張することで実現可能である。RTの構成要素 Dest, Next, Dist に加えて新たに経路固定フラグ (Flg) と有効期間 (Time) を導入する。以下にパケット送信時と RT 生成時の動作を示す。

7.1.1 パケット送信

TCPパケットの宛先ノードを Dest とする経路を参照し送信する。このとき、Flg が False であれば True にし、Time を設定する。Flg が True の場合は、Time を延長する。

7.1.2 RT生成

まず、Flg が False である経路と、Flg が True の中で有効期間を過ぎている経路を削除する。これにより、通信中の経路のみが保持される。次に、最新のトラフィック情報をもとに、すべての宛先について経路探索を行う。これは、経路探索処理の中で経路途中のノードへの経路が必要になるためである。このとき、経路は一時的に保持する RT (以下 RT バッファ) に保存する。RT バッファの情報を、RT に追加する。このとき、既に記録されている経路は上書きしない。また、追加される経路の Flg は False とする。これにより、通信中の経路は保持し、それ以外の経路は最新のトラフィック情報をもとに生成される。

7.2 パケットの種類による経路制御切替

提案方式では、UDP と TCP の通信特性の違いをルーティングに反映させるため、別々の RT を生成している。そのため、受信したパケットの種類により参照する RT を切り替える

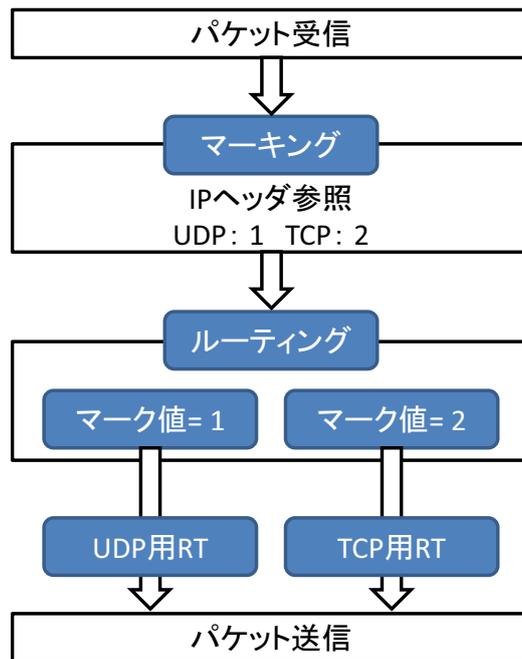


図 7.1 RT 切替フロー

必要がある。図 7.1 に、ノードがデータパケットを転送する場合の、パケット受信から RT 切替までの処理フローを示す。パケットを受信した際、IP ヘッダの情報からパケットの種類を識別する。図の例では、UDP であれば 1 を、TCP であれば 2 をマークする場合を示している。ルーティング部分では、マーク値によって参照するテーブルを切り替える。例では、マーク値が 1 ならば UDP 用 RT を、2 ならば TCP 用 RT を参照する。このような手順を踏むことで、パケットの種類による RT 切替が可能となる。

また、上記の方式においても OLSR と比較してスループットが低下する場合は、TCP 用の RT のみ OLSR と同様の動作によって生成することも手法として考えられる。この場合でも、UDP 用とは別の RT を生成することで、特性の違いを十分に活かすことができる。

7.3 リンクメトリックの拡張

本論文では、提案方式のリンクメトリックをノードのトラフィック情報および 1 ホップ分のコストによって求めているが、他にも隣接ノード数、バッテリー残量、移動速度、送信待ちパケット数といった様々な要素が考えられる。これらは、適用するネットワーク環境において優先度（重み）を設定する必要がある。例えば、移動するモバイルノードを想定した MANET (Mobile Ad-hoc Network) や車両をノードとする VANET (Vehicle Ad-hoc Network)、センサノードによる情報収集を目的としたセンサネットワークなどが考えられる。また、同じネットワークの形態であっても、使用目的によって信頼性、長寿命、低遅延など重要となる要素が異なる可能性があるため、これらについても考慮する必要がある。そ

して、そのネットワークにおいてTCPとUDP、それぞれで異なるアプリケーションの使用が想定される場合、提案方式を用いることにより異なる方針を用いた経路制御を行うことが可能となる。

第8章 まとめ

OLSR を拡張することにより，TCP 用と UDP 用の RT を別々に生成し特性の違いを活かすとともに，経路上の通信状態を考慮した経路を生成できるアドホックルーティングプロトコル PD-OLSR を提案した．TCP 用と UDP 用の RT 生成機能をシミュレータに実装し，シミュレーション評価を行った．その結果，TCP においてはパケットドロップ数を抑制することができたが，通信中に経路を切り替えることが原因となり，スループットが低下した．また，UDP においてはトラフィックの高い経路を避けた通信が行われることにより，パケット到達率が最大で約 30% 向上した．

今後は，RT の切替機能を実装し，TCP と UDP の混在環境での評価を行う．また，端末をランダムに配置した場合や，端末が移動する場合などの様々な環境での評価や，環境に合わせた新たなメトリックを検討する．

謝辞

本研究に関して、研究の方向や進め方など終始御熱心な御指導とご教示を賜りました、名城大学大学院理工学研究科情報工学専攻 渡邊晃教授に心より厚く御礼申し上げます。

本論文を作成するにあたり、快く査読を引き受けてくださり、熱心にご指導を頂きました、名城大学大学院理工学研究科情報工学専攻 山本修身教授に心より厚く御礼申し上げます。

本研究を進めるにあたり、研究内容に関して終始御熱心な御指導とご教示を賜りました、名城大学大学院理工学研究科情報工学専攻 旭健作助教に心より厚く御礼申し上げます。

本研究を進めるにあたり、研究内容に関して終始御熱心な御指導とご教示を賜りました、名城大学大学院理工学研究科情報工学専攻 鈴木秀和助教に心より厚く御礼申し上げます。

最後に、本研究を行うにあたり、適切なお検討を頂いた、名城大学工学部情報工学科渡邊研究室並びに鈴木研究室の皆様にご心より感謝致します。

参考文献

- [1] MetroMesh: <http://www.tropos.com/>.
- [2] MeshCruzer: <http://www.thinktube.com/>.
- [3] Packethop: <http://www.packethop.com/>.
- [4] Y.Amir, C.Danilov, M.: Fast Handoff for Seamless Wireless Mesh Networks, *ACM MobiSys*, pp. 83–95 (2006).
- [5] V.Navda, A.Kashyap, S.: Design and Evaluation of iMesh: An Infrastructure-Mode Wireless Mesh Network, *World of Wireless Mobile and Multimedia Networks*, pp. 164–170 (2005).
- [6] T. Clausen, E.: Optimized Link State Routing Protocol (OLSR), RFC 3626, IETF (2003).
- [7] Johnson, D.: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728, IETF (2007).
- [8] Perkins, C.: Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, IETF (2003).
- [9] Ogier, R.: Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), RFC 3684, IETF (2004).
- [10] Haas, Z. J., Pearlman, M. R. and Samar, P.: The Zone Routing Protocol (ZRP) for Ad Hoc Networks, Internet-draft, IETF, <http://tools.ietf.org/html/draft-zone-routing-protocol-00.txt> (2002).
- [11] Perkins, C. E. and Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *ACM SIGCOMM Computer Communication Review*, Vol. 24, No. 4, pp. 234–244 (1994).
- [12] V.Park and S.Corson: Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification, Internet-draft, IETF, <http://tools.ietf.org/id/draft-ietf-manet-tora-spec-04.txt> (2001).
- [13] Couto, D. S. J. D., Aguayo, D., Chambers, B. A. and Morris, R.: Performance of Multihop Wireless Networks: Shortest Path is Not Enough, *ACM SIGCOMM Computer Communication Review*, Vol. 33, No. 1, pp. 83–88 (2003).
- [14] Toh, C.-K.: Associativity-Based Routing for Ad Hoc Mobile Networks, *Wireless Personal Communications: An International Journal*, Vol. 4, No. 2, pp. 103–139 (1997).

- [15] 高橋ひとみ, 斉藤匡人, 間 博人, 戸辺義人, 徳田英幸: MANET における TCP スループット推定による経路選択機構の実環境評価, 情報処理学会論文誌, Vol. 46, No. 12, pp. 2857–2870 (2005).
- [16] Royer, E.M., C.-K. T.: A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks, *IEEE Personal Communications*, (1999)., pp. 46–55 (1990).
- [17] Lang, D.: A comprehensive overview about selected Ad Hoc Networking Routing Protocols, *Technical report, TUM*, Vol. Department of Computer Science (2003).
- [18] 間瀬憲一, 阪田史郎: アドホック・メッシュネットワーク-アドホックネットワーク社会の実現に向けて, コロナ社 (2007).
- [19] C-K.Toh: アドホックモバイルワイヤレスネットワーク-プロトコルとシステム, Vol. 共立出版, p. 2003.
- [20] Dijkstra, E.: A note on two problems in connexion with graphs, *Numerische Mathematk*, Vol. 1, No. 1, pp. 269–271 (1959).
- [21] The Network Simulator - ns-2: <http://www.isi.edu/nsnam/ns/>.
- [22] UM-OLSR: <http://masimum.inf.um.es/fjrm/development/um-olsr/>.

研究業績

学術論文

なし

国際会議

1. Yuta Mikamo, Kensaku Asahi, Hidekazu Suzuki, Akira Watanabe, "Proposal for an Ad-hoc Routing Protocol considering Traffic Conditions and Evaluation of UDP using a Redundant Route", The seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU2014), pp72-73 ,Jan. 2014.

研究会・大会等

1. 三嶋勇太, 旭健作, 渡邊晃, "トラフィック状態を考慮したアドホックルーティングプロトコルの提案と評価, 平成 24 年度電気関係学会東海支部連合大会論文集 ,Sep.2011 .
2. 三嶋勇太, 旭健作, 鈴木秀和, 渡邊晃, "通信状態を考慮した経路選択を可能にするアドホックネットワークプロトコルの提案", 情報処理学会第 74 回全国大会講演論文集, Mar.2012 .
3. 三嶋勇太, 旭健作, 鈴木秀和, 渡邊晃, "通信状態を考慮したアドホックルーティングプロトコルの提案と冗長経路に関する検討", マルチメディア, 分散, 協調とモバイル (DICOMO2012) シンポジウム論文集, Vol.2012 , No.1 , pp.1697-1703 , Jul.2012 .
4. 三嶋勇太, 旭健作, 鈴木秀和, 渡邊晃, "アドホックネットワークにおける通信状態を考慮したルーティング手法の提案", 信学技報 112(241) , pp1-6 , Oct.2012 .
5. 三嶋勇太, 旭健作, 鈴木秀和, 渡邊晃, "アドホックネットワークにおけるプロトコルごとのリンクメトリックによるルーティング手法の提案", 情報学ワークショップ 2012 (WiNF2012) 論文集, WiNF2012 , Vol.2012 , pp.201-206 , Dec.2012 .
6. 三嶋勇太, 旭健作, 鈴木秀和, 渡邊晃, "通信状態を考慮したアドホックルーティングプロトコルの提案", マルチメディア, 分散, 協調とモバイル (DICOMO2013) シンポジウム論文集, Vol.2013 , No.1 , pp.1563-1568 , Jul.2013 .

7. 三鴨勇太, 旭健作, 鈴木秀和, 渡邊晃, “PD-OLSR におけるプロトコルごとの経路制御に関する検討”, 情報処理学会研究報告, 2013-MBL-69, No.6, pp.1-6, Dec.2013 .