

# 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

三嶋 勇太 旭 健作 鈴木 秀和 渡邊 晃

名城大学大学院理工学研究科

## Proposal of Routing in Ad-hoc Networks Considering Link Metrics for Each Protocol

Yuta Mikamo Kensaku Asahi Hidekazu Suzuki Akira Watanabe

Graduate School of Science and Technology, Meijo University

### 1. 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からネットワークへの接続方法として需要が高まってきている。それに伴い、無線端末が直接通信し、自律的にネットワークを構成するアドホックネットワークに関する研究が注目を集めている。

アドホックネットワークの経路を生成するには、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を生成する必要がある。アドホックルーティングプロトコルは、IETF (Internet Engineering Task Force) において、現在まで多くの方式が標準化されている[1-7]。しかし、これらの方式は、経路生成の際に中継ホップ数が最短となる経路（最短経路）を探索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は実装に任されている場合が多い。そのため、トラフィックが集中した中継ノードが発生すると、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[8]。

複数経路の中から、適切な経路を選択することを目的としたアドホックルーティングプロトコルの研究として、以下のものが挙げられる。ABR (Associativity-Based Long-lived Routing) [9]の経路選択では、リンク切断が長時間起らない、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では、ビーコンの受信回数に差が出ないため、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) [10]は DSR (Dynamic Source Routing Protocol) [4]を拡張することにより、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 スループットは所定のモデル式を使って計算される。モデル式には遅延 (RTT: Round-Trip Time) と往復パケット喪失率 (RTPL: 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 スループットだけに着目しており、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り、ネットワークのオーバーヘッドが高くなるという課題がある。

IP ネットワークでは、フロー制御やウィンドウサイズを変化させる輻輳制御を行い、ネットワーク帯域を有効に使用しようとする TCP と、輻輳制御を行わず、端末が意図した通

信がそのまま送出される UDP という異なる性質の通信が存在する。負荷分散や通信の安定化の観点からマルチパス通信も含めた様々なプロトコルが提案されているが、UDP と TCP の性質の違いを考慮し経路を分けるような経路選択方式は存在しない。本稿では、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プロトコル OLSR (Optimized Link State Routing) を拡張することにより、ルーティングテーブル (以下 RT と記述) を TCP 用と UDP 用で別々に生成し、TCP と UDP の通信特性を生かした最適な経路選択を可能とする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。

以下、2 章で OLSR を例にして既存技術の問題点を示し、3 章では PD-OLSR の概要、4 章でシミュレータ上での実装について示す。5 章で動作検証、6 章で評価を行い、7 章でまとめる。

### 2. 既存技術の課題

既存のアドホックルーティングプロトコルの例として OLSR の RT 生成動作および IP ネットワークにおける UDP と TCP の特性について示し、既存技術における問題点を挙げる。

#### 2.1 OLSR

OLSR は常時経路を生成しておく Proactive 型のルーティングプロトコルである。OLSR では各ノードが隣接ノードへ定期的にブロードキャストする HELLO と、ネットワーク全体へ定期的にフラッディングする TC という制御メッセージを送受信することにより、自身の存在をネットワークの全ノードに把握させる。HELLO と TC で送信される情報は、各メッセージの送信元ノードのアドレス、送信元ノードが把握している自身の隣接ノードのアドレス、情報の新しさを識別するシーケンス番号などである。これらの情報は RT を生成するために必要な情報であり、各ノード内の情報リポジトリに登録される。OLSR の RT 生成プロセスは HELLO と TC の受信により、情報リポジトリが更新されていくことによって進行する。

#### 2.2 UDP と TCP の特性の違い

TCP/IP ネットワークでは UDP と TCP という特性の異なる通信が存在する。UDP では端末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、ネットワーク内のパケットロスの影響が考慮されることはない。これに対し TCP では輻輳制御の機能により ACK が順調に返ってく

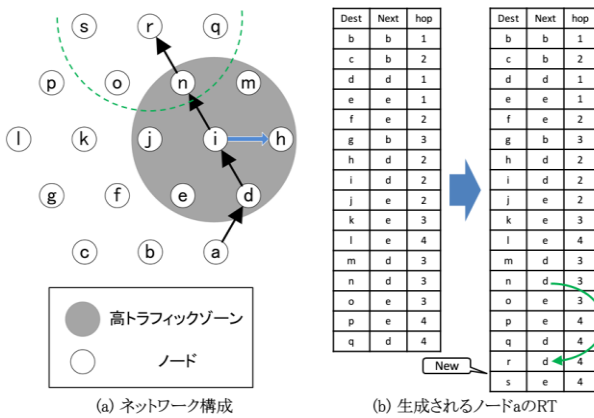


図 1 OLSR の RT 生成

るとウィンドウサイズを拡大し、帯域を有効に使うとする。パケットロスを検出するとネットワークの輻輳が発生したものと判断し、ウィンドウサイズを縮小する。このようにウィンドウサイズが適切に調整され、ネットワークの更なる輻輳を防止する。TCP と UDP が混在したネットワークにおいては、ネットワーク上のトラフィックはまず送出された UDP パケットの合計により UDP が占めるトラフィックが定まり、残りの帯域を複数の TCP セッションが分け合う形になる。

図 2 に UDP について、図 3 に TCP についてのシミュレーションで求めたマルチホップ通信における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が可能な距離だけ離して一直線上に並べホップ数を 1~10 ホップで変化させた場合のスループットを測定した。UDP では、一般にネットワーク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。これに対し、TCP では輻輳制御によってネットワーク帯域を使い切ろうとするため、ホップ数の増加とともにスループットが大きく低下する。

このように、UDP と TCP ではノードから送出されるトラフィック量およびマルチホップ通信時のホップ数によるスループットの変化が大きく異なる。

既存のルーティングプロトコルでは前述した OLSR の動作のように、ホップ数のみを基準に経路を生成するものが多い。OLSR では、複数の最短経路が存在する場合、どの経路を選択するかという手順は定義されていないため、実際に生成される経路は、実装に任されており、多くの場合、最初に発見された最短経路が選択される。ここで図 1(a) に示すネットワークにおいて、ノード i から h への通信がすでに行われていたものとする。ノード i から送信されるキャリアは隣接ノード e, d, h, j, m, n にも届く。そのため仮に図 1 (a) のような経路が生成されると、ノード i の周辺はトラフィックが増加し、スループットが低下する可能性がある。このように OLSR ではネットワークのトラフィックに偏りがあった場合、最適な経路を生成することができない。

さらに、既存技術では特性の異なる UDP と TCP の通信を同一の RT を用いて制御を行っており、通信特性の違いを経路生成に反映することができない。

### 3. PD-OLSR

#### 3.1 概要

PD-OLSR は、UDP/TCP のそれぞれで、通信状態を計測し、

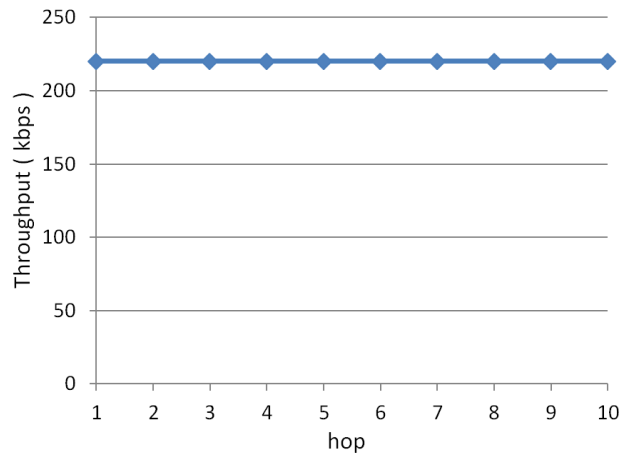


図 2 マルチホップ通信における UDP スループット

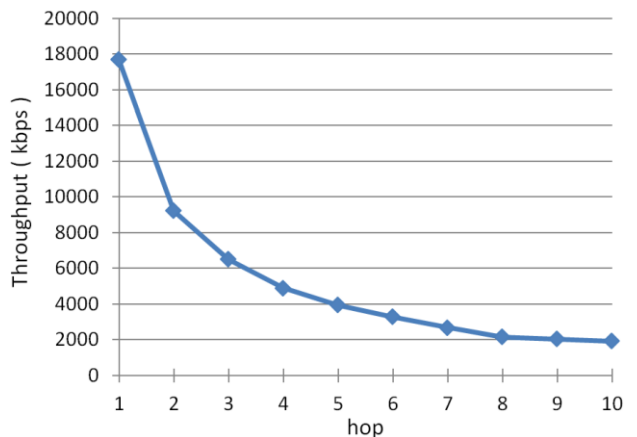


図 3 マルチホップ通信における TCP スループット

その情報をもとに経路を生成する。UDP ではホップ数が増加してもスループットは変化しない。そのため、最短経路よりもホップ数を伸ばした冗長経路を選択することを許容できると考えられる。経路生成のアルゴリズムとしてダイクストラ法を用いることにより、取り得るすべての経路の中から最適な経路を生成する。以下に、UDP/TCP それぞれの経路選択指標、ダイクストラ法の適用、および経路生成手法について示す。

#### 3.2 PD-OLSR の通信状態指標

PD-OLSR では OLSR の基本部分はそのまま利用し、制御メッセージに各ノードが測定した自身の通信状態を表す情報を OLSR の制御メッセージに追加して送受信する。受信したノードはその情報を元に UDP と TCP それぞれ専用の RT を生成する。そのため、UDP と TCP の経路選択に用いる指標を別々に考える。UDP の経路選択指標は UDP トラフィック (UDP Traffic)、TCP の経路選択指標は TCP セッション数 (TCP Session) とする。UDP トラフィックとは各ノードが検出した UDP によるキャリアの総量で、TCP セッション数は各ノードが検出する TCP セッション数の合計である。

#### 3.3 ダイクストラ法の適用

PD-OLSR では経路探索を行う際に、ダイクストラ法[11]を用いる。ダイクストラ法は、グラフ上の 2 頂点間の最短経路を効率的に求めるアルゴリズムである。経路の合計コストを基準に経路を求めることができる。

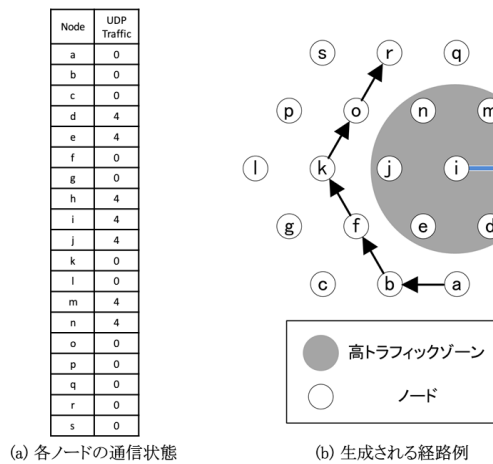


図 4 PD-OLSR で生成される経路例

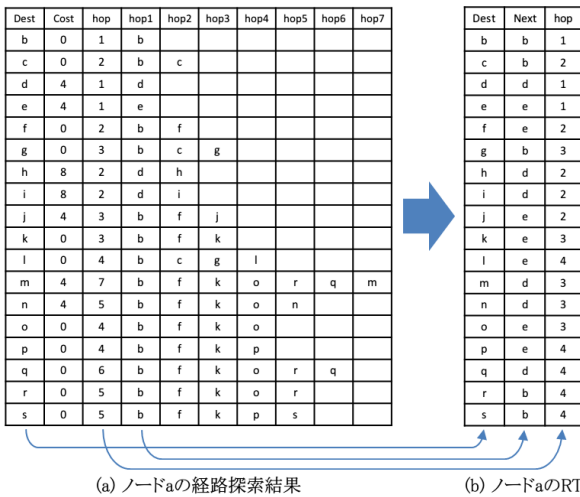


図 5 PD-OLSR の RT 生成

既存の OLSR では、ネットワークボロジの情報から最短経路を得ている。それに対し、ダイクストラ法を用い、経路コストとして各ノードが測定した通信状態を使用することにより、例えば UDP における経路上の総トラフィック量を基準に経路選択が可能となる。この方式によると、ホップ数が多いが混雑している部分を迂回した経路を選択することが可能である。

3.4 PD-OLSR の経路生成手法

3.4.1 生成される経路例

図 4 に PD-OLSR で生成される経路例、図 5 に RT 生成の様子を示す。これらの図を用いて UDP 用の RT 生成を例にして、PD-OLSR の経路生成手順を示す。TCP 用の RT 生成についても、コスト計算方法は異なるが、UDP 用 RT の生成手順と同様の方法で実現できる。トラフィックの条件は図 1 の場合と同じである。図 4 (a) のテーブルは、各ノードが計測した UDP Traffic の情報 (個数) である。ノード i からノード h への通信が行われているため、隣接ノードであるノード d, e, j, m, n では UDP Traffic が検出されている。ここで検出されるトラフィックを仮に 4 として記載している。

3.4.2 経路生成時の動作

PD-OLSR では、各ノードが計算した自身の通信状態を表

す UDP Traffic 情報を HELLO メッセージと TC メッセージに載せて隣接ノードへ広告する。各ノードは制御メッセージによって共有したネットワーク内のノードの隣接ノード情報と UDP Traffic 情報からリンクメトリックに変換し、その情報を元にダイクストラ法によって各宛先に対して、複数存在する経路の中から最適な経路の探索を行う。経路探索結果は図 5 (a) のようになり宛先ノード (Dest)、探索された経路のコスト (Cost)、ホップ数 (hop) および経路中の中継ノード (hop1, hop2, ...) から構成される。ここで、リンクメトリックへの変換は、両端ノードの UDP Traffic の和をリンクメトリックとすることにより行う。

変換したリンクメトリックをもとに経路探索を行う。経路探索は、取り得る経路すべてを探索し、すべての経路について経路中のリンクメトリックの合計値が最小のものを選択する。経路コストが最小のものが複数存在する場合、その中でホップ数が最小のものを、さらにそれも複数存在する場合には先に探索されたものが選択される。PD-OLSR の RT は各宛先について探索された最適経路の次ホップおよびホップ数を、RT に追加していくことにより生成する。

生成される経路は、各 Dest に対して経路の合計コストが最小のものである。例えば、図 4 でノード a からノード r へ UDP で通信が行われると、高トラフィックゾーンを避けた経路 [a→b→f→k→o→r] が生成される。

TCP 用の RT 生成については、上記の説明において、UDP Traffic を TCP Session 数をコストに置き換えたもので計算することにより生成できる。ただし、TCP ではホップ数が増加すると大きくスループットが低下するため、最短経路の中から最適な経路を選択するものとする。TCP 用の RT 生成の場合、各 Dest に対して最短経路の中から経路中の TCP Session が最小となる経路が選択される。もし、TCP Session が同じであった場合は、UDP Traffic の少ない経路が選択される。

4. シミュレータへの実装

PD-OLSR をネットワークシミュレータ ns-2[12] に実装した。以下に UDP 用の RT 生成機能を例にして、その概要を示す。

4.1 ns-2 の変更部分

図 6 に、ns-2 の内部構造と変更部分を示す。MAC 層に PD-OLSR の UDP Traffic を計測するモジュールを追加した。また、UDP Traffic 計測モジュールで計測した UDP Traffic をルーティングエージェントで呼び出せるようにし、ルーティングエージェントの OLSR を PD-OLSR の経路生成動作が行えるよう拡張した。

4.2 OLSR の拡張

OLSR において、制御メッセージと情報リポジトリには図 7 で示すような関係がある。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。さらに、隣接ノード集合、2 ホップ隣接ノード集合、トポロジ集合の情報を元に RT を生成する。

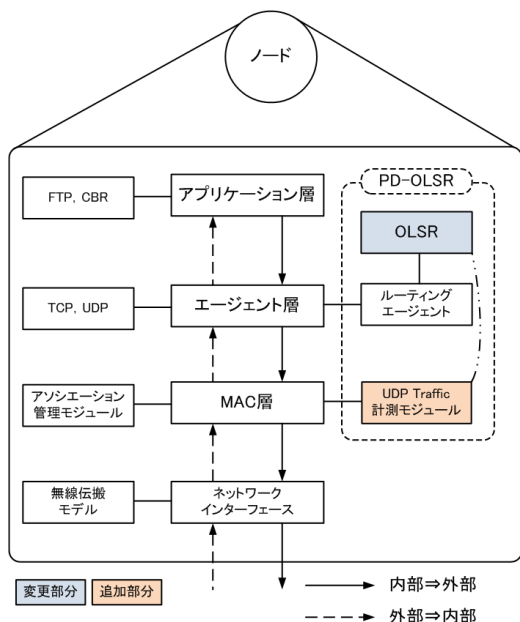


図 6 ns-2 の内部構造と変更部分

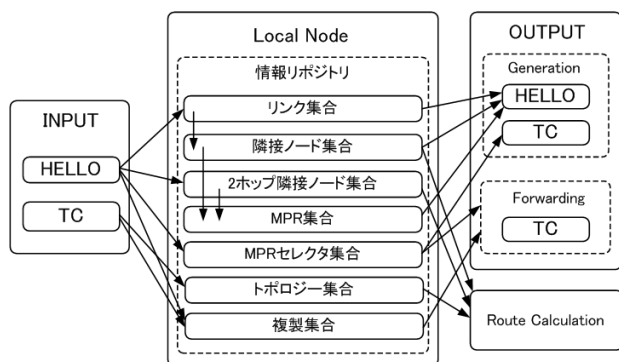


図 7 制御メッセージとリポジトリの関係

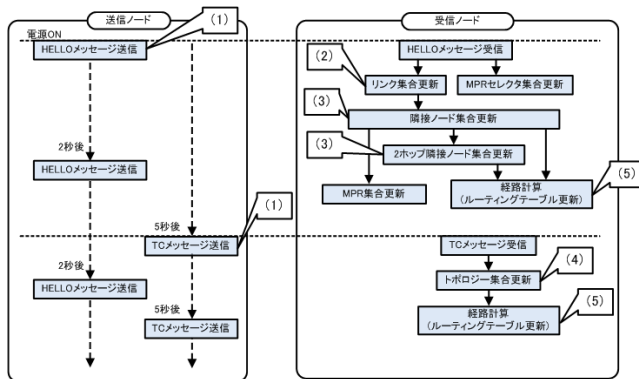


図 8 OLSR の改造箇所

OLSR の改造において、図 7 の情報リポジトリ内のリンク集合、隣接ノード集合、ホップ隣接ノード集合、トポロジ集合に通信状態情報である UDP Traffic の情報を追加した。

OLSR の送信ノードと受信ノードにおける制御メッセージの処理の流れを図 8 に示す。以下に拡張したそれぞれの処理を示す。

- (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 Traffic を付加
- (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記録に送信元ノードの通信状態情報を記録
  - 一致する記録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記録を生成
- (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記録に通信状態情報を記録
- (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記録に通信状態指標を記録
  - 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記録を生成
- (5) 経路計算
  - 3 章に示す方法で RT を生成

### 5. 動作検証と拡張

UDP 用 RT の生成機能を実装し、動作検証を行った。以下にその内容を示す。

#### 5.1 検証内容

ノード 19 台を規則的に配置、電波到達範囲を隣接ノードまでとしアドホックネットワークを構築した。ノードの持つ RT が安定した後、ノード b から r へ UDP 通信を行い、その通信によるトラフィックが制御メッセージによって更新されることにより、経路がどのように切り替わるか確認を行った。

#### 5.2 検証結果

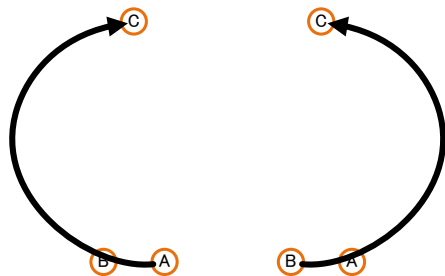
通信開始時、最短経路のうちの 1 つが選択され通信が行われた。その後、通信によるトラフィック検出、制御メッセージによるトポロジ情報の通知・更新が行われると、更新された情報をもとに経路を生成する。この動作を制御メッセージの受信ごとにくり返し、最新の通信状態をもとに経路を切り替わる様子を確認した。このことから、UDP 通信用の RT 生成機能が正しく実装されていることがわかった。

しかし、経路が切り替わる中で、経路ループによって通信ができない時間帯が発生した。経路ループは隣り合うノード同士でパケットを投げ合う形で発生し、制御メッセージによってトポロジ情報が更新されると解消される。経路ループの発生要因として、図 9 に示すような同一の宛先に対し、例えばノード A がノード B を、ノード B がノード A をという形でお互いに次ホップとして登録し合うことにより、パケットを送り合うループに陥ることが挙げられる。今回の検証においてもこの事象を確認した。すなわち、通信負荷の高いノードを避け、迂回した経路を生成する際に、隣り合うノードで逆方向に迂回する経路を生成したことにより、経路ループに陥った。

制御メッセージによって情報を伝達しているため、ノードの持つ情報を同期することは困難である。そこで提案方式では逆方向に迂回する経路を生成しても、互いに次ホップとして指定し合うことを防ぐ手法を追加した。

迂回経路について、迂回する度合いが大きいと、逆方向に迂回する経路があったときに経路ループに陥りやすい。迂回





(a) ノードAのノードCへの経路 (b) ノードBのノードCへの経路

図 9 経路ループ発生の要因

の度合いを少なくすることによりこれを回避できると考えられる．そこで，経路の迂回の度合いを押さえることにより経路ループを抑制するよう拡張を行った．

5.3 PD-OLSR の拡張

PD-OLSR では，経路コストを基準に経路を選択する．経路コストにホップ数に起因するコストを導入することにより，経路のホップ数増加，つまり迂回の度合いを調整することができる．リンクメトリックはリンクの両端ノードのトラフィックの和としており， $i$  ホップ目のリンクメトリックを  $M_i$ ， $1$  ホップ分のコストを  $H$  としたとき  $n$  ホップの経路コスト  $C$  は，

$$C = \sum_{i=0}^n M_i + nH \tag{1}$$

と表される．本拡張では， $H$  をネットワーク全体のトラフィック量に合わせて変動させるため，ネットワーク全体のノードのトラフィック量の最大値  $T_{max}$  と係数  $\alpha$  との積とする．ノード数  $m$  台のネットワークにおいて，それぞれのノードのトラフィックを  $T_1, T_2, \dots, T_m$  とするとき， $T_{max}$  は次の式で求めるものとする．

$$T_{max} = \frac{1}{m} \sum_{i=1}^m T_i \tag{2}$$

$T_{max}$  を用いると  $n$  ホップの経路コスト  $C$  は，

$$C = \sum_{i=0}^n M_i + n\alpha T_{max} \tag{3}$$

となる．

リンクメトリックにホップ数に当たるコストを導入することにより，ネットワークが空いている状態では迂回する経路を，混雑してくるとともに迂回を抑制し最短経路またはそれに近い経路を選択する．また， $\alpha$  の値を大きくすることにより，迂回度合いの制限が大きくなり，また一定以上値にすると，最短経路の中から一番コストの小さい経路を選ぶことになる．

6. 評価

図 1(a) の構成を拡大し，ノード数を 37 台としたときの大規模なシミュレーションによる評価を行った．UDP 通信は VoIP を想定し，ネットワークに高負荷を与えた場合に，PD-OLSR のパケットロスに関する影響を調べた．

6.1 環境

シミュレーション環境を表 1 に，ノード配置を図 10 に示す．シミュレーションの開始から終了までの時間を 300 秒とし，

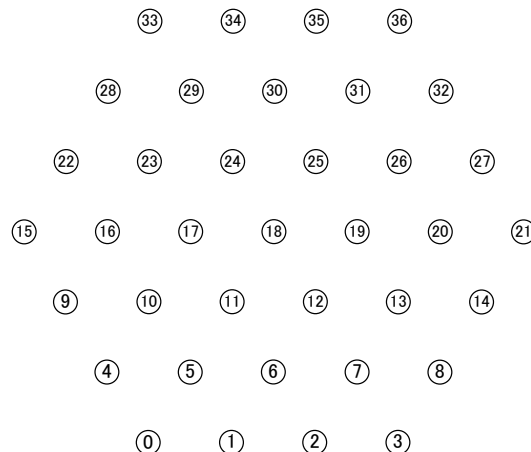


図 10 評価シミュレーションのノード配置

表 1 シミュレーション環境

通信環境	
通信規格	IEEE802.11g
ノード数	37 台
電波到達範囲	隣接ノード
通信組	2 台 1 ペア
通信ノード選択手法	ランダム
通信パラメータ	
通信タイプ	CBR
トランスポートプロトコル	UDP
パケットサイズ	200[Byte]
パケット発生率	64[kbps]

シミュレーション開始 30 秒後から 10 秒間隔で UDP セッションを増加させた．通信を行うノードの組み合わせはランダムに選定した．上記のシミュレーションを 4 回行い，平均を求めた．

6.2 結果

すべての UDP セッションの送信元ノードの送信パケット数の合計と，宛先ノードの受信パケット数の合計からネットワーク全体のパケットロス数を求め，OLSR と  $\alpha$  の値を変化させたときの PD-OLSR を比較した．図 11 に OLSR と PD-OLSR で式 2 の  $\alpha$  を 0.5~3.0 としたときのシミュレーション結果を表 2 それぞれのパケットロスの改善率の比較を示す．PD-OLSR(0.5) は， $\alpha=0.5$  の場合の PD-OLSR の結果を表している．

今回測定した  $\alpha=0.5\sim3.0$  範囲では，すべての場合において，OLSR よりもパケットロスが小さい結果となり，その改善率の最大値は 41.2% となった．また， $\alpha$  の値が 2.5 と 3.0 の 2 つの場合の結果が同じ値になることから，今回のシミュレーション環境では  $\alpha$  を 2.5 以上としたときに最短経路の中から最適な経路を選択していると考えられる．

$\alpha=2.0$  の結果を見ると， $\alpha=2.5$  および 3.0 の場合よりもわずかによい結果が得られた．これは，必ずしも最短経路が最適な経路ではないことを示している．このときどの程度迂回させるのが適切かは，ネットワークトポロジに依存すると考えており，さらにノード数を増やした場合等様々なトポロジでの検証を行っていく必要がある．

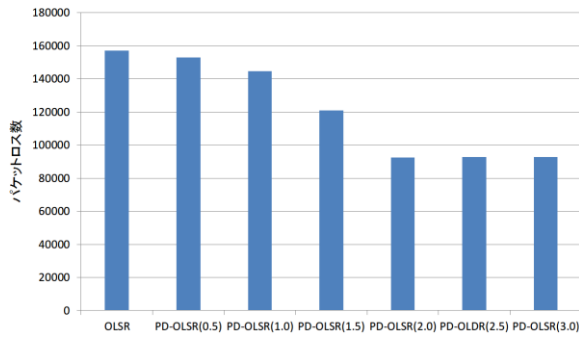


図 11 シミュレーションによるパケットロスの比較

表 2 パケットロスと改善率の比較

$\alpha$	OLSR	PD-OLSR					
	—	0.5	1.0	1.5	2.0	2.5	3.0
パケットロス数	157193	153006	144776	121107	92435	92681	92681
パケットロス改善率	—	2.7%	7.9%	23.0%	41.2%	41.0%	41.0%

## 7. 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, 経路上の通信状態を考慮して経路を生成できるプロトコル PD-OLSR を提案した. RT を分けることによりそれぞれの通信特性に合わせた経路選択を行う. リンクメトリックをもとにダイクストラ法による経路探索を行うことにより, 最短経路によらず最適な経路を選択可能である.

UDP 通信用の RT 生成機能をシミュレータに実装し, 動作検証を行った. 動作検証により経路ループの発生を確認したため, その対策として経路コストにホップ数の情報を含め, ホップ数をのばしすぎないようにすることにより, 迂回度合いを調整する.

シミュレーションで, PD-OLSR において迂回度合いを決定する係数を変化させた場合と OLSR との比較評価を行い, パケットロスを最大で約 41%改善することができた.

今後は, PD-OLSR の実装を完了し, UDP と TCP が混在する環境といった様々な環境でのシミュレーションを行い PD-OLSR の効果を確認する. また, 経路の迂回係数に関する検証も行う.

## 参考文献

- [1] T. Clausen, Ed.: Optimized Link State Routing Protocol (OLSR), RFC3626 (2003)
- [2] D. Johnson: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC4728 (2007)
- [3] C. Perkins: Ad hoc On-Demand Distance Vector (AODV) Routing, RFC3561 (2003)
- [4] R. Ogier: Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), RFC3684, IETF (2004).
- [5] Zygmunt J. Haas, Marc R. Pearlman, Prince Samar: The Zone Routing Protocol (ZRP) for Ad Hoc Networks, Internet draft, IETF MANET Working Group (2002). Expiration: January, 2003.
- [6] Charles E. Perkins, Pravin Bhagwat: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, ACM SIGCOMM, Vol. 24, No. 4 (1994).
- [7] V. Park, S. Corson: Temp orally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification, Internet draft, IETF MANET Working Group (2001).
- [8] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, Robert Morris: Performance of Multihop Wireless Networks: Shortest Path is Not Enough, ACM SIG-COMM
- [9] Toh, C.-K.: Associativity-Based Routing for Ad-Hoc Mobile Networks, Wireless Personal Communications, Vol. 4, No. 2, pp. 103-139 (1997).
- [10] 高橋ひとみ, 齊藤匡人, 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).
- [11] Dijkstra, E. W. (1959). A note on two problems in Connexion with Graphs. In Numerische Mathematik, 1 (1959), S. 269~271.
- [12] The Network Simulator - ns-2: <http://www.isi.edu/nsnam/ns/>.

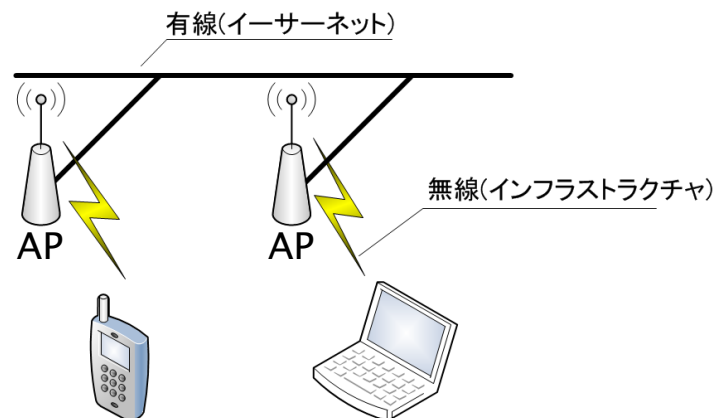


# アドホックネットワークにおけるプロトコルごとの リンクメトリックによるルーティング手法の提案

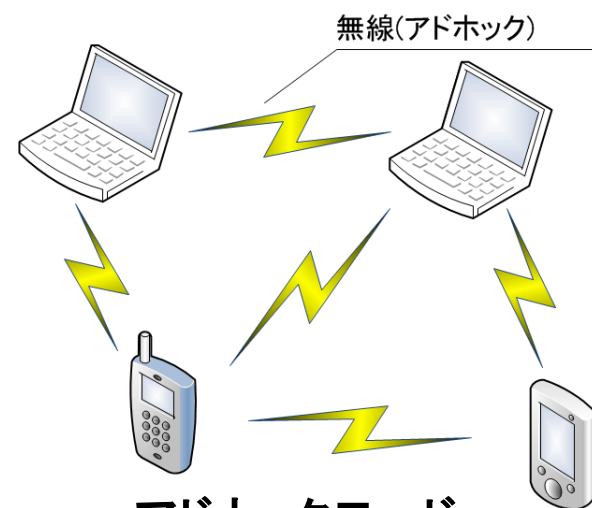
名城大学大学院 理工学研究科  
三嶋 勇太 旭 健作 鈴木 秀和 渡邊 晃

# 研究背景

- ▶ 無線LANの普及
  - スマートフォン, タブレットの普及
- ▶ インフラストラクチャモード
  - AP(Access Point)を中継点として各端末が通信を行う通信方式
  - 現在普及している形
- ▶ アドホックモード
  - 中継装置を介さず各端末が直接通信を行う通信方式



インフラストラクチャモード



アドホックモード  
(アドホックネットワーク)



# アドホックネットワーク

- ▶ アドホックルーティングプロトコル
  - アドホックネットワークに特化したルーティングプロトコル
  - 周辺ノードと制御メッセージをやりとりしRT(Routing Table)を生成
- ▶ 利用形態
  - インフラを利用できない環境
  - 災害時, イベント会場など一時的な通信
  - 無線メッシュネットワーク
    - アクセスポイント同士がアドホックネットワークで接続

# 既存方式:OLSR

# OLSRの概要 (Optimized Link State Routing)

- ▶ プロアクティブ型
  - 通信要求発生前からRTを生成しておく
- ▶ 周辺ノードの情報を収集することによってRTを生成
  - 各ノードは定期的に制御メッセージを送受信
- ▶ 制御メッセージ
  - HELLOメッセージ
    - 各ノードが持つ情報を通知
    - 2秒毎に隣接ノードへブロードキャスト
  - TCメッセージ
    - ネットワークトポロジーを通知
    - 5秒毎にネットワーク全体にフラッディング

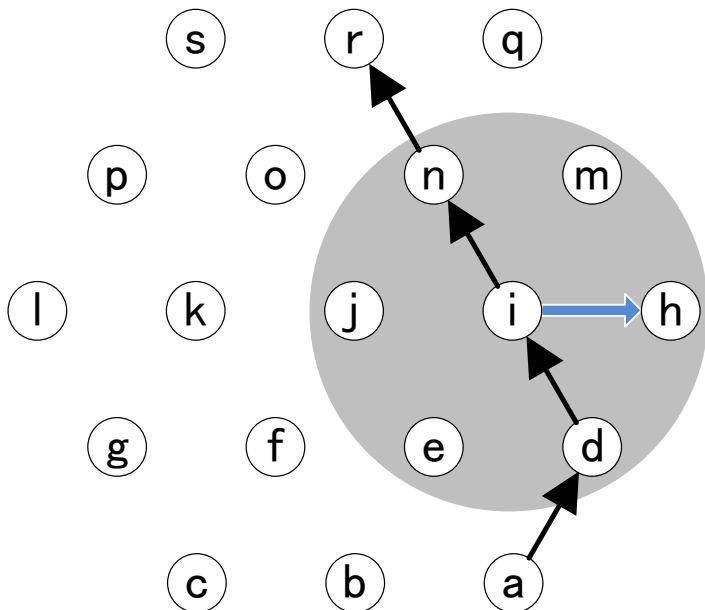
制御メッセージには  
リンク情報のみ

# OLSRのRT

Dest:宛先ノード  
 Next:次ホップノード  
 hop:宛先ノードまでのホップ数

- ▶ 制御メッセージのやりとりによってRTが生成されていく

Dest	Next	hop
b	b	1
d	d	1



ノード数 : 19台  
 電波到達範囲: 隣接ノードまで  
 既に行われている通信: i→h

OLSRで生成される経路例

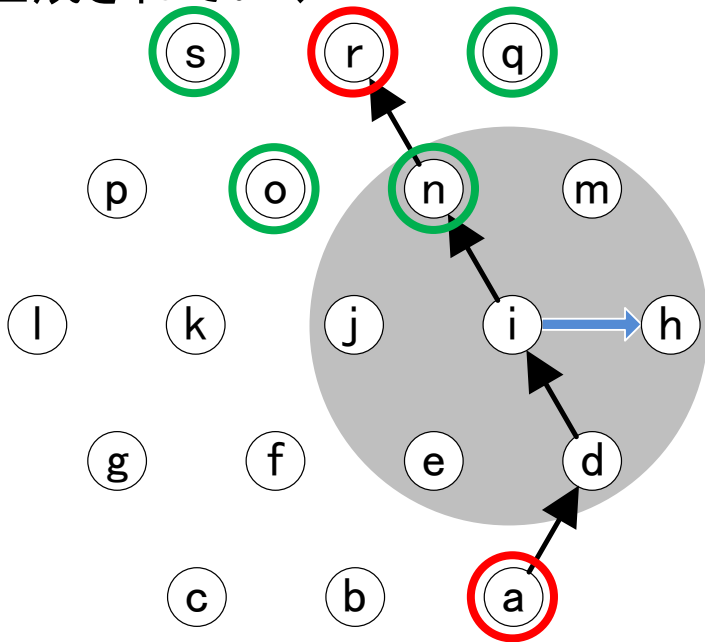
● 高トラフィックゾーン  
 ○ ノード

j	e	2
g	b	3
k	e	3
m	d	3
n	d	3
o	e	3
l	e	4
p	e	4
q	d	4

# OLSRのRT

Dest: 宛先ノード  
 Next: 次ホップノード  
 hop: 宛先ノードまでのホップ数

▶ 制御メッセージのやりとりによってRTが生成されていく



OLSRで生成される経路例

● 高トラフィックゾーン  
 ○ ノード

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

New



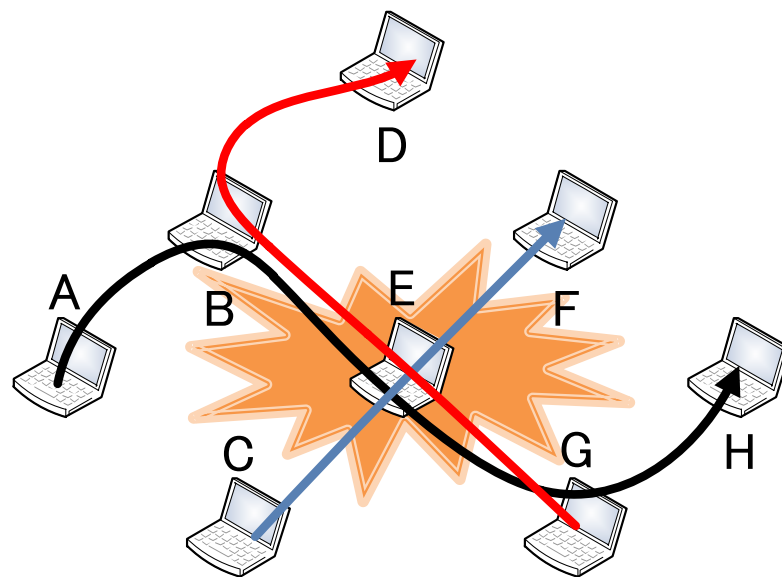
# 課題1

- ▶ 経路の中継ホップ数が最小となる経路を選択
- ▶ 最短ホップ数の複数の経路の中からどの経路が選ばれるかは定義されていない
- ▶ 複数の通信で同一のノードを経由する経路が選択され、トラフィックが集中する可能性も

パケットロスが多発



スループットが低下



# UDPとTCPの特性

## ▶ UDP通信

- 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

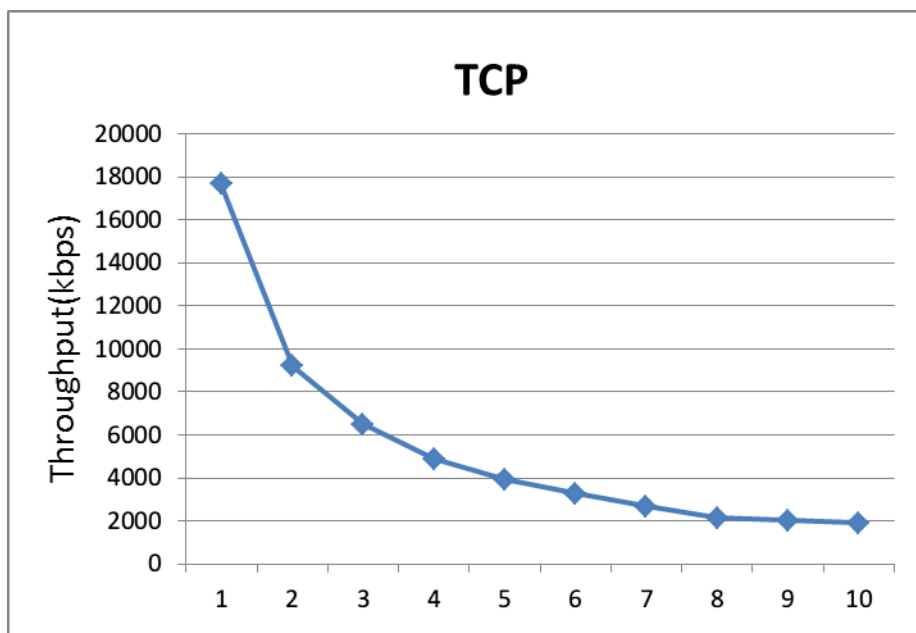
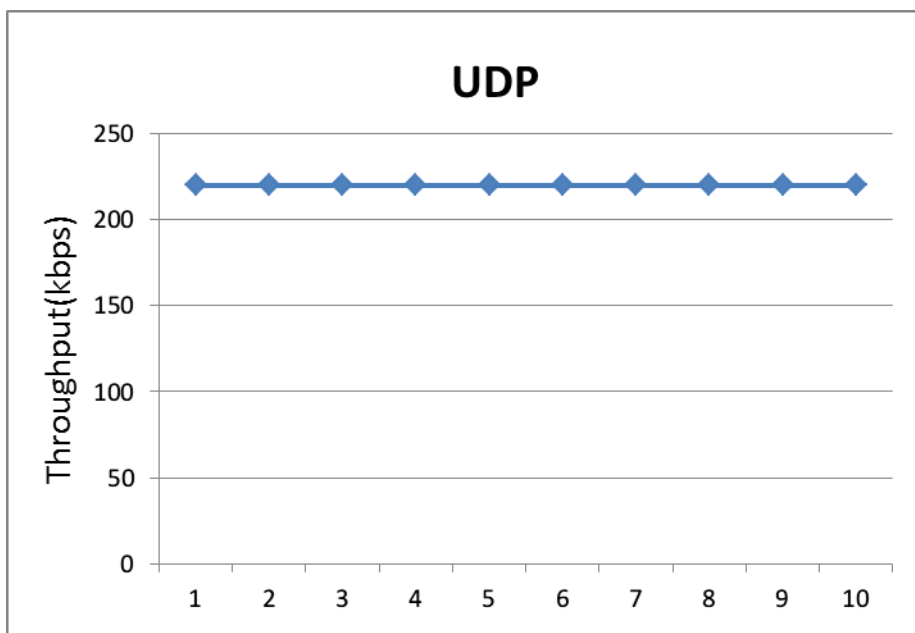
## ▶ TCP通信

- 輻輳制御によって順調にACKが返ってきた場合はウィンドウサイズを拡大し帯域を使いきろうとする

UDP通信とTCP通信が混在するネットワークのトラフィックは、送出されるUDPパケットの合計からUDPが占めるトラフィック量が定まり、残りの余裕のある帯域分を複数のTCPセッションが分け合う

# UDPとTCPのマルチホップ通信

- ▶ 1～10ホップのスループットをシミュレーションで測定
  - ノードを一直線上に配置



TCPのスループットはホップ数に反比例して低下

## 課題2

- ▶ UDPとTCPでは通信性質が異なる
  - トラフィック発生量
  - ホップ数によるスループット変化
- ▶ 既存のルーティングプロトコルでは2種類の通信は同一RTを使用



同一経路を用いることによって  
TCP通信のスループットが低下する可能性

# 課題

- (1) 最短経路が複数存在するときの選択基準がない
- (2) UDP/TCPで同一のRTを用いて制御



# 提案方式:PD-OLSR

# 提案方式: PD-OLSR (Protocol Dependent-OLSR)

- ▶ OLSRを改造
- ▶ 目的
  - UDP用とTCP用別々にRTを生成
  - トラフィックの高いノードを避けた経路選択

# PD-OLSR 通信状態指標

- ▶ UDP通信 : UDP Traffic
  - 自身が検出するネットワーク上のキャリアの総量
- ▶ TCP通信 : TCP Session
  - 自身が検出するTCPセッション数の合計
  - 各セッションがウィンドウサイズを増減
  - セッション間の公平を保つため
- ▶ 各ノードが計測

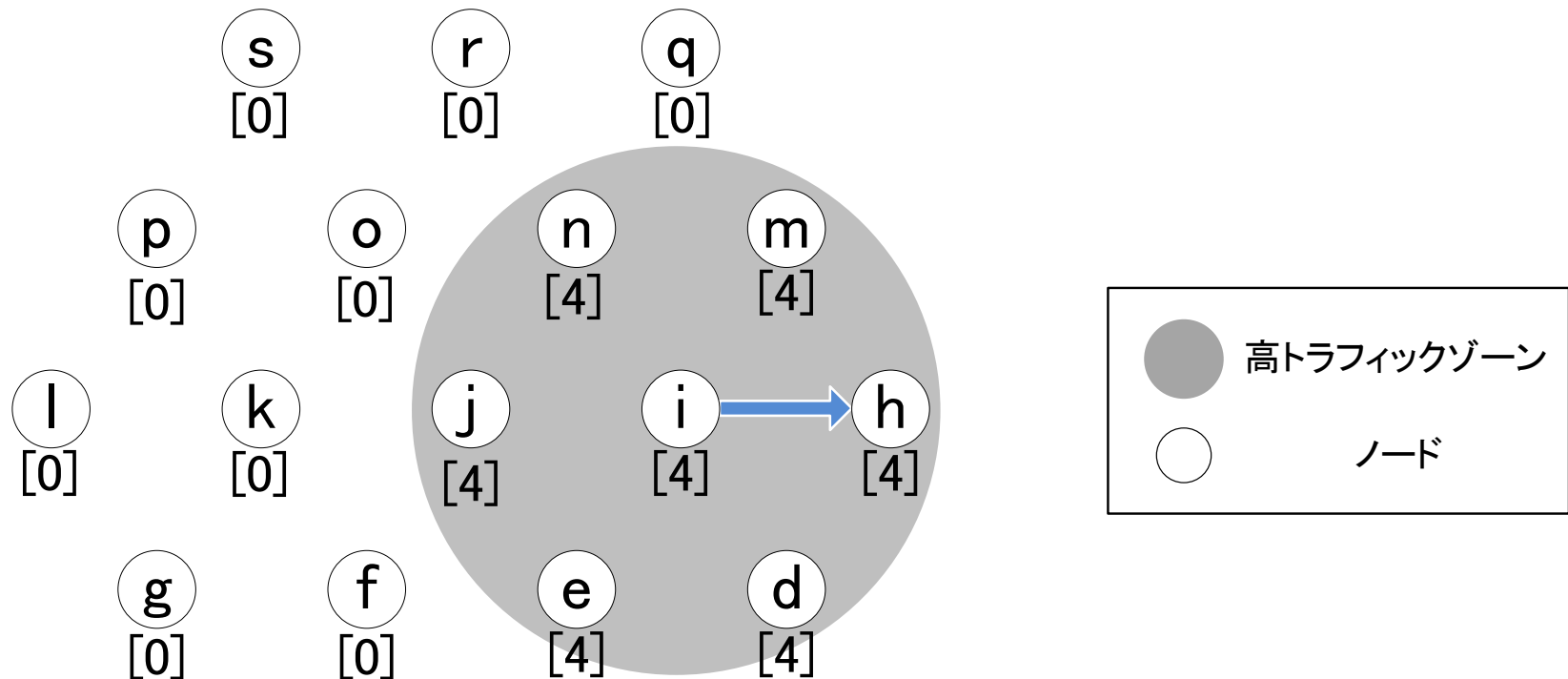
# 制御メッセージ

- ▶ OLSRの制御メッセージの仕組みはそのまま用いる
- ▶ トラフィックの情報を制御メッセージ(HELLO,TC)に追加,  
ネットワーク全体に広告
- ▶ ネットワーク内のノードのトラフィック情報を共有

# 動作



# 環境



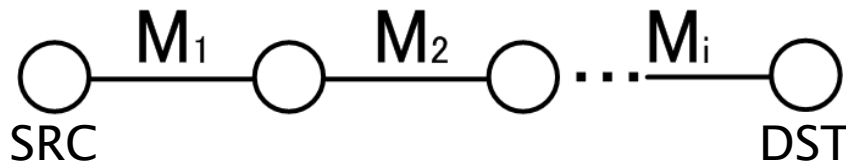
- ▶ [ ]内の数字は各ノードのトラフィック
- ▶ i→hの通信のトラフィック量:4

# 経路探索(1/3)

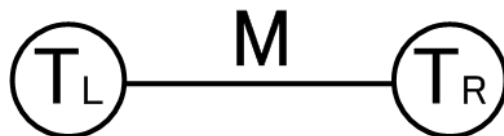
- ▶ 制御メッセージによって共有した情報をもとに行う
- ▶ アルゴリズム:ダイクストラ法
  - グラフ理論における最短経路問題解決アルゴリズム
- ▶ UDP/TCPの経路選択基準
  - UDP:ホップ数によってスループットが低下せず  
ホップ数を増やした経路も許容できる  
→ すべての経路から最適なものを選択
  - TCP:ホップ数に反比例してスループット低下  
→ 最短経路の中から最適なものを選択

# 経路探索(2/3)

- ▶ 経路コストが最小のものを選択
  - コストC : 経路中のリンクメトリックの総和
  - リンクメトリックM : リンク両端ノードのトラフィックの和  
 $T_L, T_R$  : リンク両端ノードのトラフィック

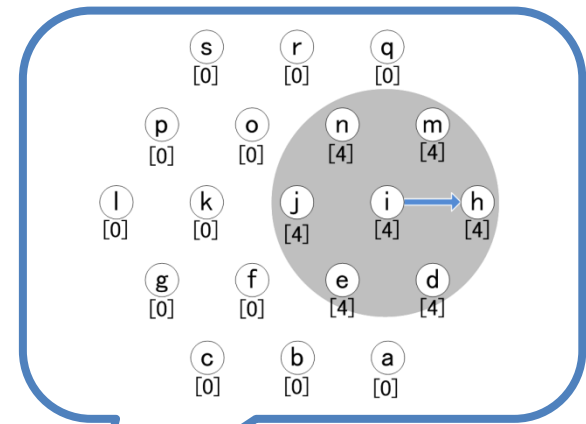
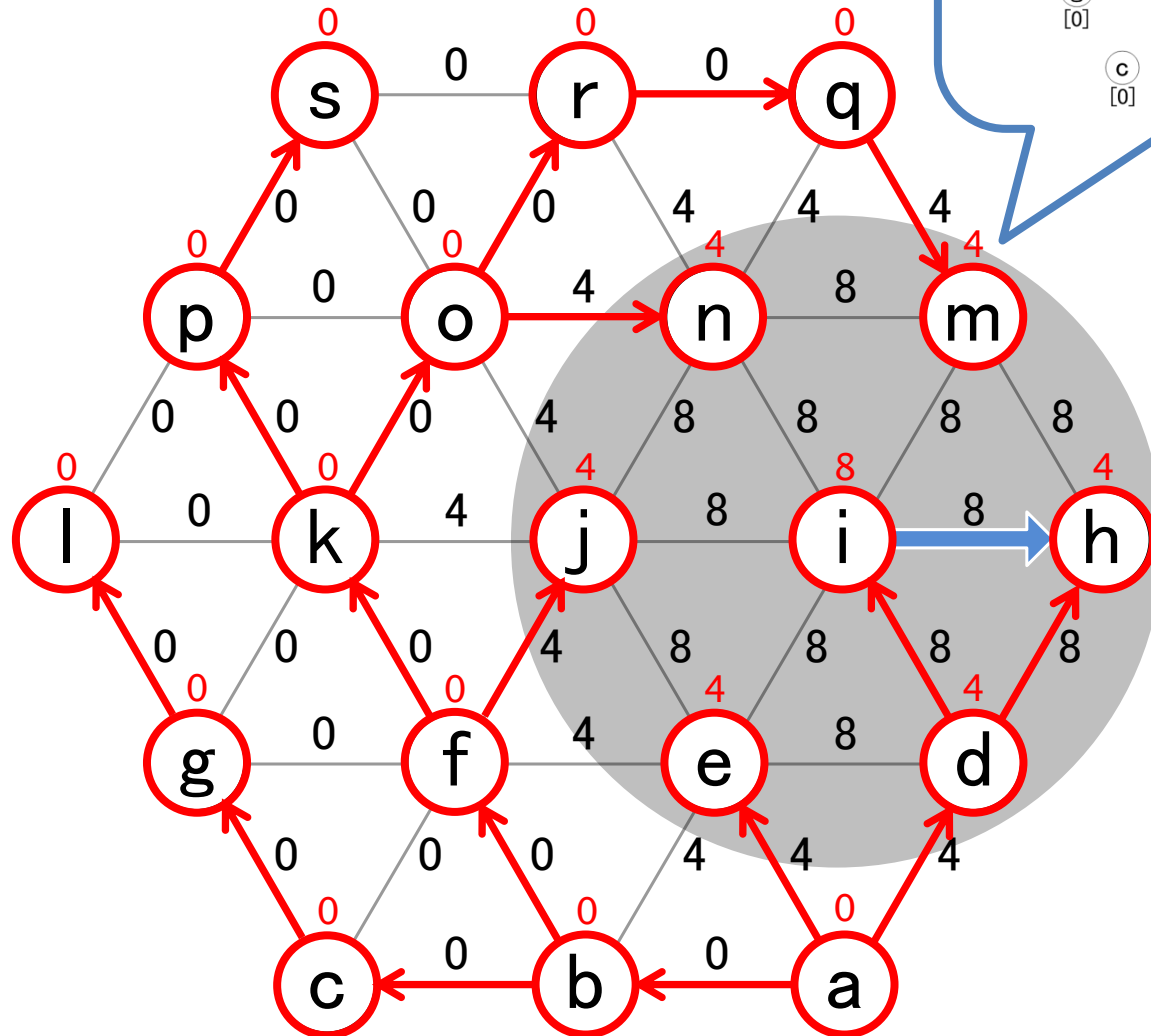


$$C = \sum_{i=1}^n M_i$$



$$M = T_L + T_R$$

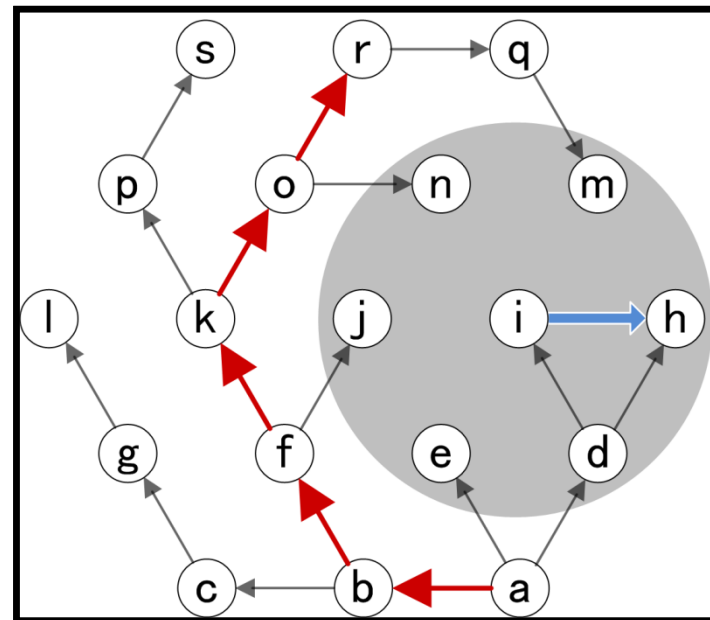
# 経路探索 (3/3)



# 経路探索結果

冗長経路を含めたすべての経路を探索し  
合計コストが一番小さいものを選択

Dest	Cost	hop	hop1	hop2	hop3	hop4	hop5	hop6	hop7
b	0	1	b						
c	0	2	b	c					
d	4	1	d						
e	4	1	e						
f	0	2	b	f					
g	0	3	b	c	g				
h	8	2	d	h					
i	8	2	d	i					
j	4	3	b	f	j				
k	0	3	b	f	k				
l	0	4	b	c	g	l			
m	4	7	b	f	k	O	r	q	m
n	4	5	b	F	k	O	n		
o	0	4	b	F	k	O			
p	0	4	b	F	k	P			
q	0	6	b	F	k	O	r	q	
r	0	5	b	f	k	O	r		
s	0	5	b	f	k	p	s		



ノードaから各ノードへの経路

ノードaによる経路探索結果

# RT生成

Dest
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s

hop	hop1
1	b
2	b
1	d
1	e
2	b
3	b
2	d
2	d
3	b
3	b
4	b
7	b
5	b
4	b
4	b
6	b
5	b
5	b

Dest → Dest  
hop → hop  
hop1 → Next  
としてRTに保存



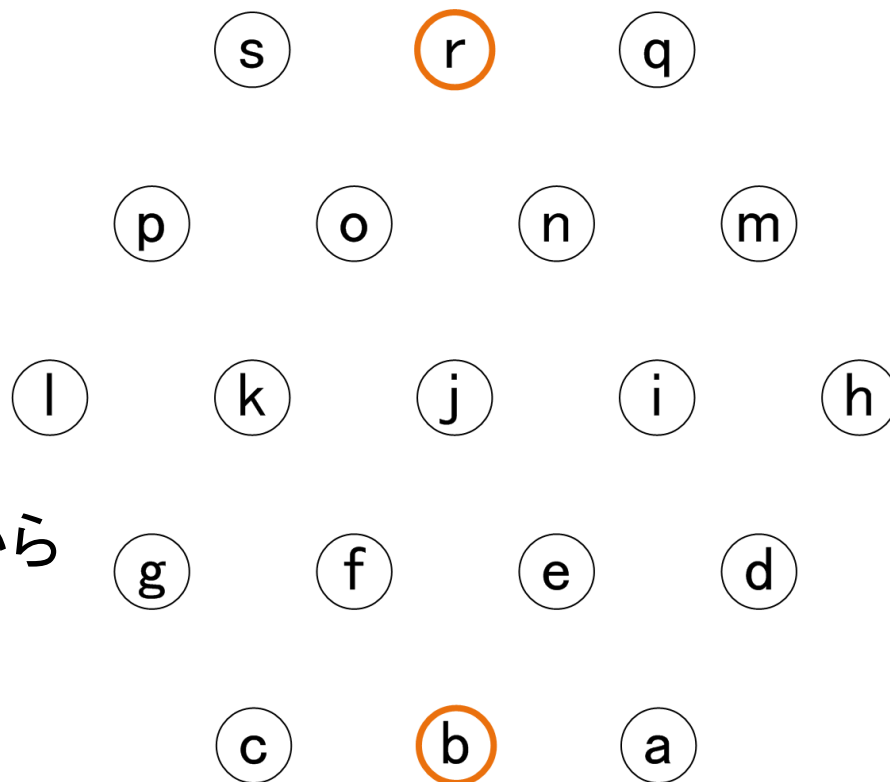
Dest	Next	hop
b	b	1
c	b	2
d	d	1
e	e	1
f	e	2
g	b	3
h	d	2
i	d	2
j	e	2
k	e	3
l	e	4
m	d	3
n	d	3
o	e	3
p	e	4
q	d	4
r	d	4
s	e	4



# 動作検証

# 環境

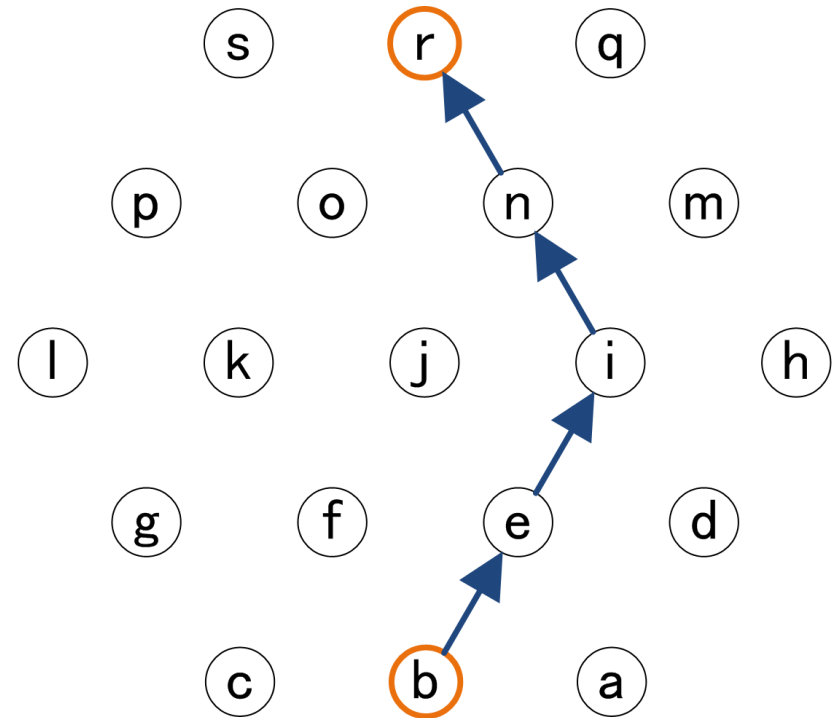
- ▶ シミュレータ: ns-2
- ▶ ノード数: 19台
- ▶ 電波到達範囲: 隣接ノード
- ▶ IEEE802.11g
- ▶ シミュレーション開始30秒後から20秒間UDP通信
- ▶ ノードb → r





# 動作検証 (1 / 3)

最短経路で通信開始



# 動作検証 (2/3)

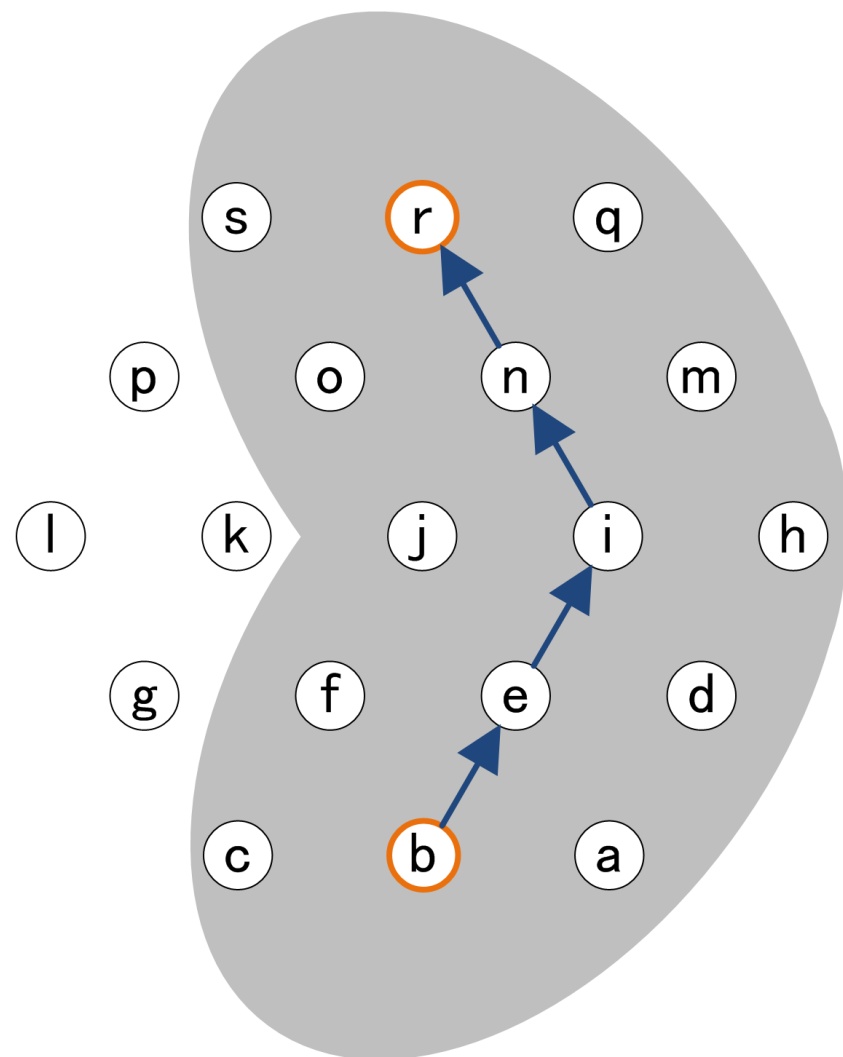
最短経路で通信開始



トラフィック検出



制御メッセージ送受信



# 動作検証 (3 / 3)

最短経路で通信開始



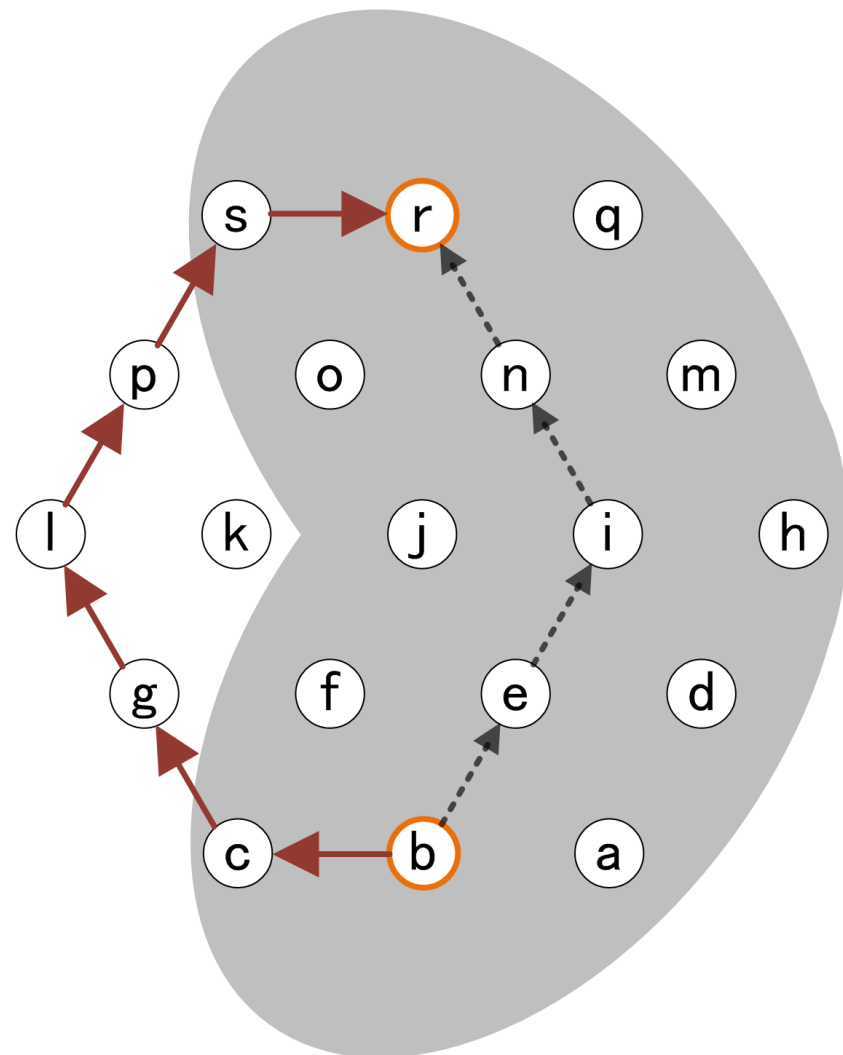
トラフィック検出



制御メッセージ送受信



経路切替



# 動作検証 (3 / 3)

最短経路で通信開始



トラフィック検出



制御メッセージ送受信

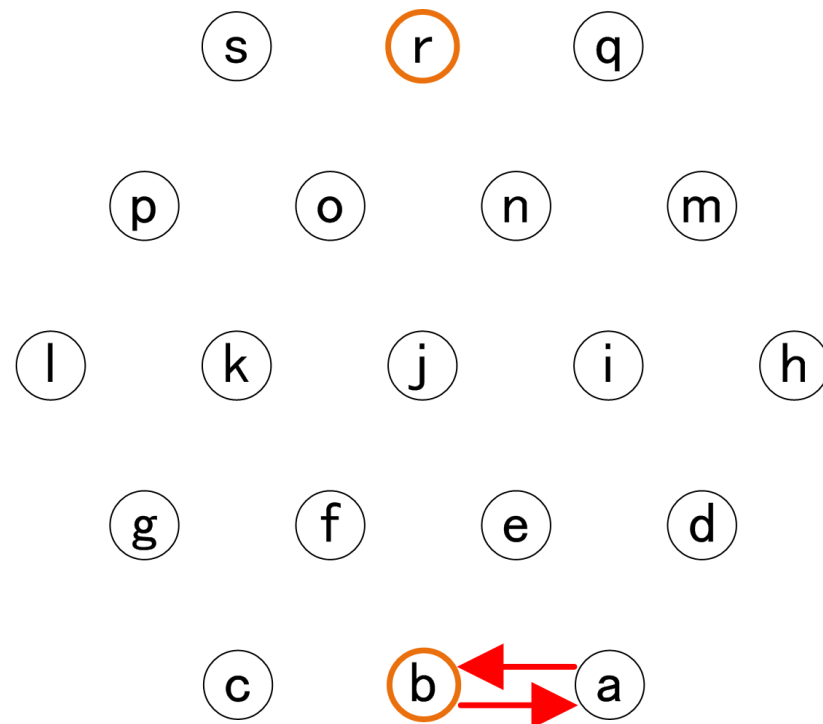


経路切替



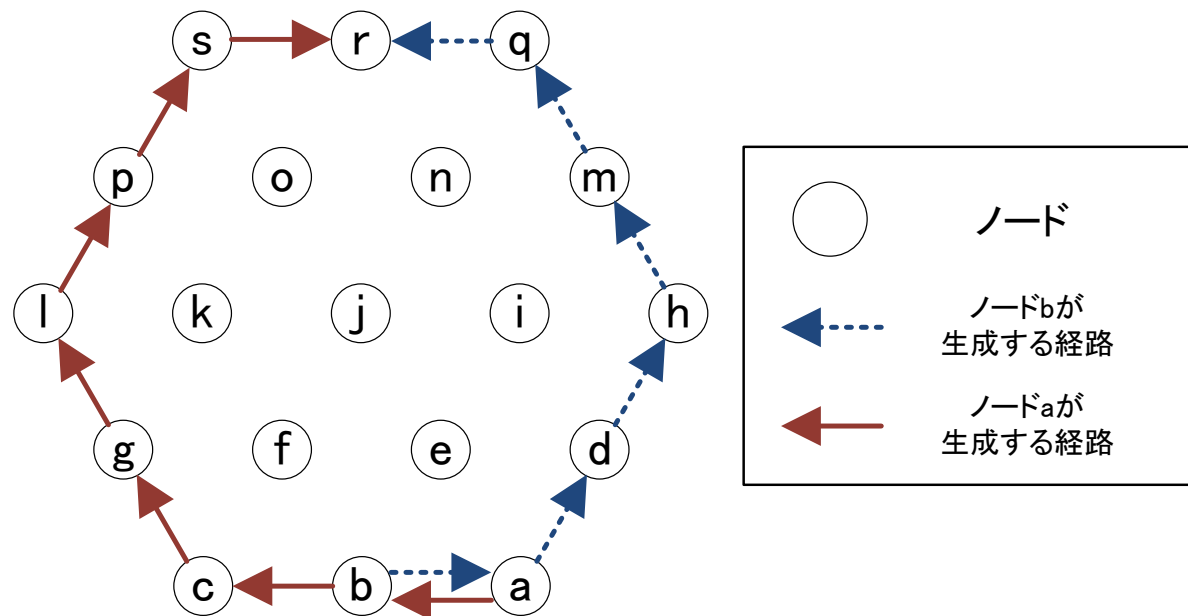
さらに経路が切り替わると...

**経路ループ発生**



# 経路ループ

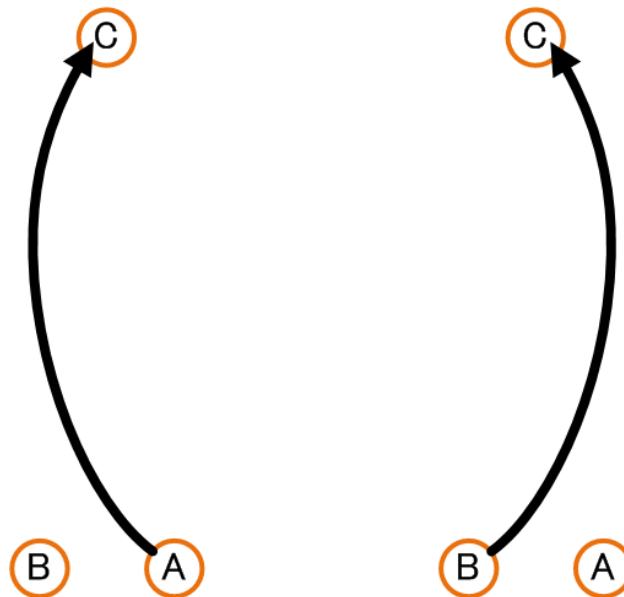
- ▶ ノードa, b間で確認
- ▶ 発生時の経路探索結果



隣接ノードで逆方向の経路を探索している

# 経路ループ対策

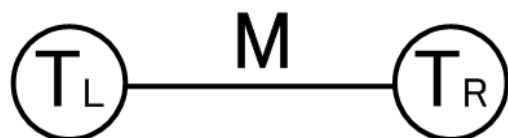
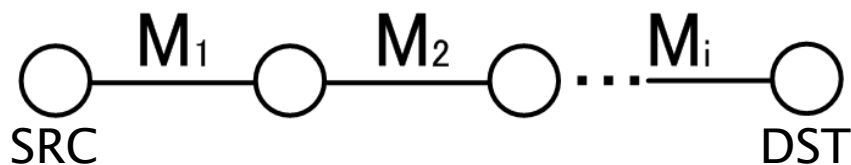
- ▶ 経路コストにホップ数に関するコストを導入
- ▶ ノードA, Bでお互いを次ホップとするCへの経路を選択しない



(a) ノードAのノードCへの経路

(b) ノードBのノードCへの経路

# リンクメトリックの拡張



$$M = T_L + T_R + \underline{\underline{\alpha T_{max}}}$$

$T_{max}$ : ネットワーク全体の  
ノードのトラフィックの最大値

$\alpha$ : 係数

経路コスト  $C = \sum_{i=1}^n M_i$

リンクメトリック  $M = T_L + T_R$

+

$$\alpha T_{max}$$

ホップ数により増加する  
経路コスト

$\alpha$  によって迂回を調整

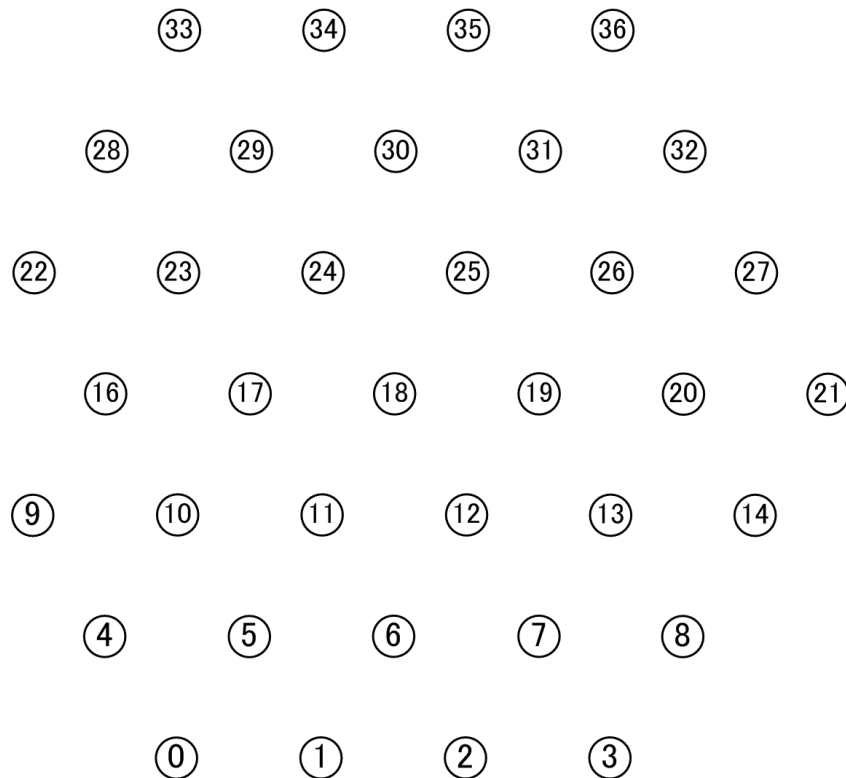
ある程度制限することで経路ループが発生しなくなることを確認

# シミュレーション評価



# 環境

無線規格	IEEE802.11g
ノード数	37[台]
通信組	2台1ペア
通信組選択方法	ランダム
通信タイプ	CBR
トランスポートプロトコル	UDP
ルーティングプロトコル	OLSR, PD-OLSR
パケットサイズ	200[Byte]
レート	64[kbps]

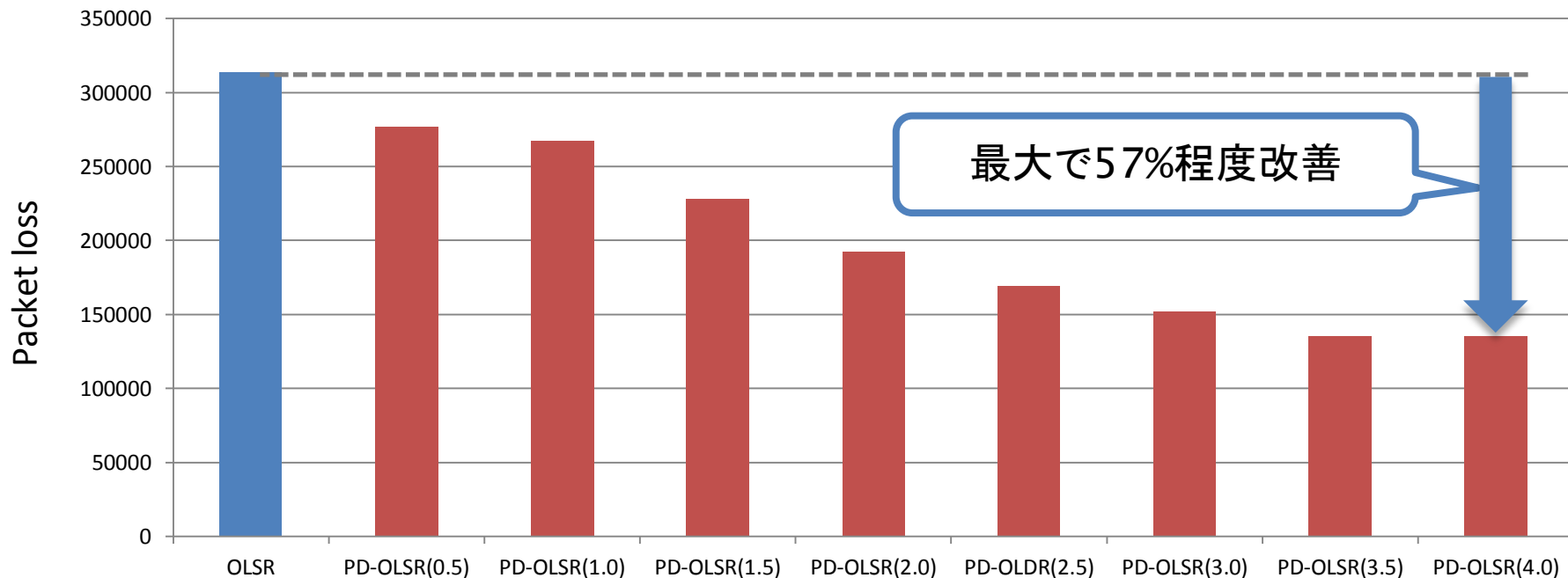


- ▶ 開始30秒後から10秒間隔でUDPセッション増加
- ▶ 計530秒

OLSRとPD-OLSRにおいて $\alpha$ を0.5~3で変化させた場合  
それぞれ10回ずつ行い、ネットワーク全体のパケットロス数の平均を比較

# 結果

$\alpha$ を増加させるとパケットロスも減少  
3.5と4.0のとき最短経路



	OLSR	PD-OLSR							
$\alpha$	—	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
パケットロス数	313847	276990	267527	227593	192273	169124	151780	135256	135256
改善率	—	11.7%	14.8%	27.5%	38.7%	46.1%	51.6%	56.9%	56.9%

# むすび

## ▶ 本発表

- OLSRを拡張することによって、UDP用とTCP用それぞれのRTを別々に生成し、経路上の通信状態を考慮して経路生成ができるプロトコルPD-OLSRを提案した
- UDP通信用RT生成機能を実装し、迂回に関するパラメータを変化させる評価シミュレーションを行った結果パケットロスが最大で57%改善されることを確認した

## ▶ 今後

- 提案方式のシミュレータへの実装を完了し、RTをUDPとTCPで分けたことによる効果を検証する
- ネットワークの規模を大きくした場合等の様々な環境での評価