

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

三嶋 勇太[†] 旭 健作[†] 渡邊 晃[†]

名城大学理工学部[†]

1. はじめに

無線 LAN を標準搭載した携帯端末の普及に伴い、無線端末のみでネットワークを構築するモバイルアドホックネットワーク (MANET: Mobile Ad-hoc Network) の研究が期待されている。MANET で提案されている多くのアドホックルーティングプロトコルは、中継端末の数が最小となるような経路 (最短経路) を選択する。しかし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く、1つの端末が複数の通信の中継を行うような経路が選択されると、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。本稿では OLSR (Optimized Link State Routing) [1] を拡張し、TCP と UDP の各プロトコルごとにトラフィックなどを考慮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。

2. OLSR

2.1 OLSR の概要

OLSR とは、MANET のルーティングプロトコルの中で、通信要求が発生する前から経路を生成しておき、要求発生時にすぐ通信を開始できるという特徴を持つプロアクティブ型のプロトコルである。HELLO、TC といった制御メッセージによってノード同士で定期的に情報をやりとりし、ネットワーク内の情報を共有する。

共有された情報は各ノードの情報リポジトリに登録され、ルーティングテーブル (以下 RT と記述) を生成する際に使用される。情報リポジトリが更新されることで RT 生成が進行する。

2.2 OLSR の RT 生成

OLSR の RT は宛先ノード (Dest), Dest への次ホップノード (Next), Dest までのホップ数 (hop) から構成され、各 Dest に対して 1 つの経路を保持する。図 1 に OLSR の

RT 生成手順と生成される経路例を示す。簡単のためノードは等間隔で配置されており電波到達範囲は隣接ノードまでとする。図 1 に示す RT は、ノード a を対象としてノード j への経路が新たに生成される様子を示している。制御メッセージの送受信によってノード j への 1 つの最短経路 [a→b→d→h→j] が完成する。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おらず、実際の経路は、実装に依存したものとなり、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、ノード d から g への通信が行われているものとする。ノード d から送信されるキャリアは隣接ノード b, e, h にも届く。そのため図 1 のような経路が生成されると、ノード d 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が経路生成時に考慮されていない。

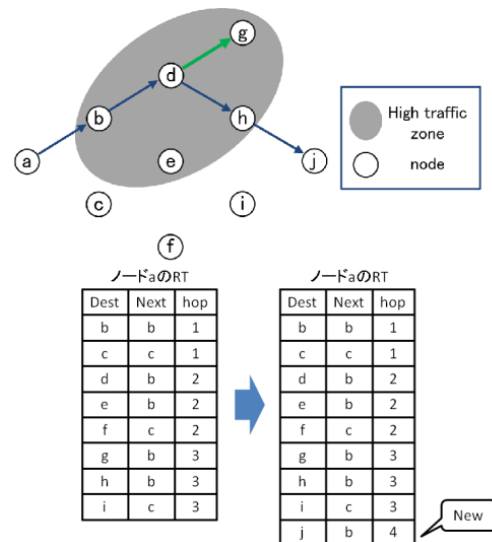


図 1 OLSR の RT 生成

3. 提案方式 PD-OLSR

3.1 TCP と UDP 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

Proposal on an Ad-hoc Routing Protocol to enable Routing considering Traffic Condition and its Evaluation

[†] Yuta Mikamo, Kensaku Asahi, Akira Watanabe
Faculty of Science and Technology, Meijo University

ーク内のパケットロスとは考慮していない。これに対し TCP では輻輳制御の機能により ACK が順調に返ってくるとウィンドウサイズを拡大し、帯域を有効に使用するとする。パケットロスが発生すると輻輳が発生したと判断し、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。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量が定まり、残りの帯域を複数の TCP セッションが分け合う。

3.2 PD-OLSR の経路選択指標

PD-OLSR では OLSR の基本部分はそのまを用い、制御メッセージに自身の通信状態を表す情報を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 通信と TCP 通信それぞれの専用の RT を生成する。3.1 の考えに基づき、UDP 通信と TCP 通信の経路選択に用いる指標を別々に考える。UDP 通信の経路選択指標は UDP トラフィック、TCP 通信の経路選択指標は TCP セッション数とする。UDP トラフィックとは各ノードが検出するネットワーク上のキャリアの総量で、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 RT の生成方法

図 2 に UDP 通信用の RT 生成を例にして、PD-OLSR の経路生成手順を説明する。トラフィックの条件は図 1 の場合と同じである。図 2 の (a) のテーブルは各ノードが計算した UDP トラフィックの情報の一覧である。ノード d から g への UDP 通信が行われているためノード d および g の隣接ノードで UDP トラフィックが検出されている。ここでは仮にこのトラフィック量を 4 としている。(b) は UDP 通信用の RT を生成するためにノード a が持つ、新た

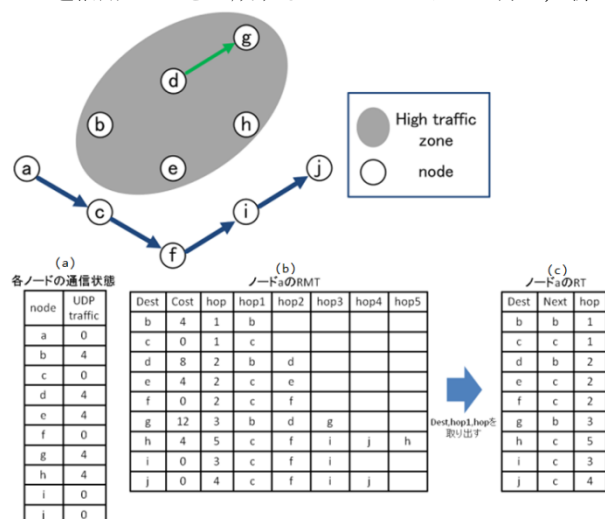


図 2 PD-OLSR の RT 生成

に定義した経路計算テーブル(RMT : Route Metric Table)である。RMT は宛先ノード(Dest), 宛先への経路の合計メトリック(Cost), ホップ数(hop), および Dest を含めた経路中のノード(hop1, hop2, ...)から構成される。各ノードはダイクストラ法によってネットワーク内のノードへの経路探索を行い、その結果を RMT に保存する。図 2(c) のテーブルは(b)をもとに生成された UDP 通信 RT である。この方法によって、最短経路だけでなくホップ数を増やした最適な経路を選択することができる。

3.4 双方向メトリック

ダイクストラ法ではリンクのメトリック(コスト)を用いている。PD-OLSR で取得できるトラフィックは、ネットワーク内の各ノードのトラフィックであるため、リンクのメトリックに変換する必要がある。変換時にリンクのメトリックは双方向で異なるものに変換する。双方向で異なるメトリックとは図 3 において a がノード A の、b がノード B のトラフィックとすると、A→B 方向のリンクメトリックとして b を、B→A 方向のリンクメトリックとして a を用いることで、メトリックの更新には宛先方向のノードのトラフィックのみが必要となる。双方向で同じメトリック、具体的には a+b を用いた場合と比較すると、経路選択能力は等しくなるが、最新のメトリックに更新するためには a, b 両方の情報が必要となる。つまり、双方向で異なるメトリックを用いることで、双方向で同じメトリックを用いる場合と比較して、メトリックの更新に必要な情報を少なくすることができる。

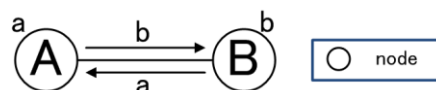


図 3 双方向で異なるメトリック

4. むすび

OLSR を拡張することで、TCP 用と UDP 用の RT を別々に生成し、ノードのトラフィック状況を考慮した経路選択が可能となるプロトコル PD-OLSR を提案した。ダイクストラ法および双方向で異なるメトリックを用いることで、冗長経路選択を可能にし、トラフィック状況を経路選択により反映しやすくする拡張を検討した。

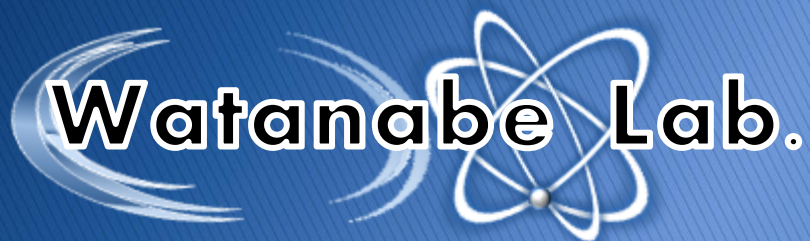
文 献

- [1] T.Clausen and P.Jacquet ,October 2003
RFC3626 (OLSR)



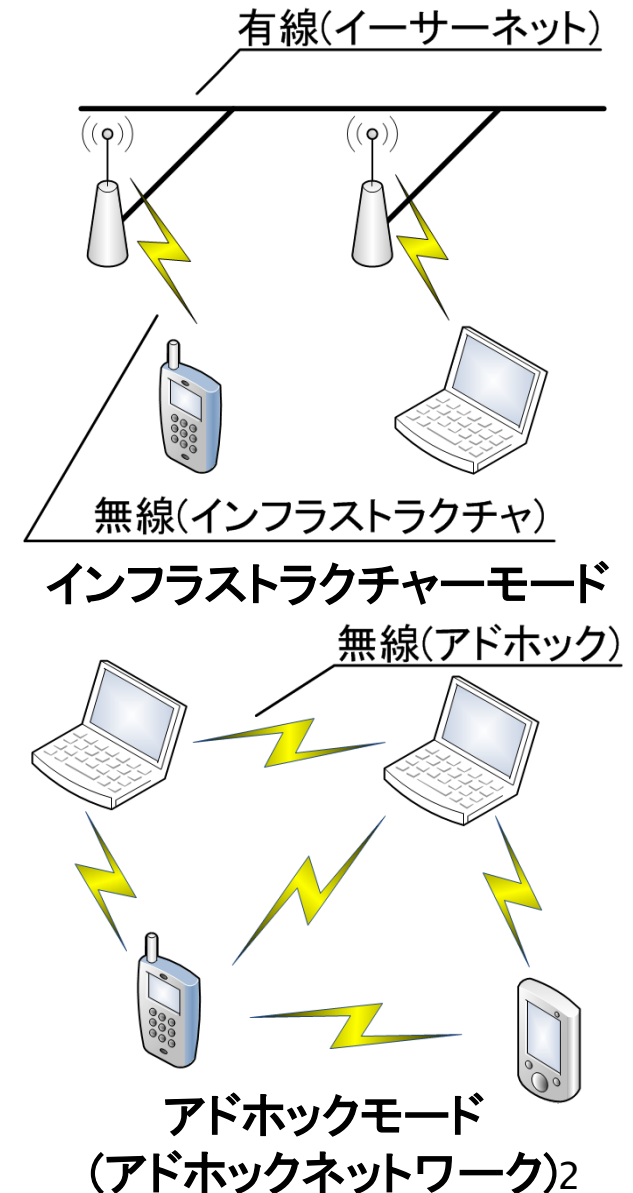
通信状態を考慮した経路選択を可能にする アドホックネットワークプロトコルの提案

名城大学 理工学部 情報工学科
三鴨勇太 旭 健作 渡邊 晃



背景

- ▶ スマートフォンやタブレット端末の影響もあり, wi-fi等の無線LANが急速に普及している
- ▶ 無線LAN
 - 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
 - アドホックモード
 - 各端末が直接通信を行う



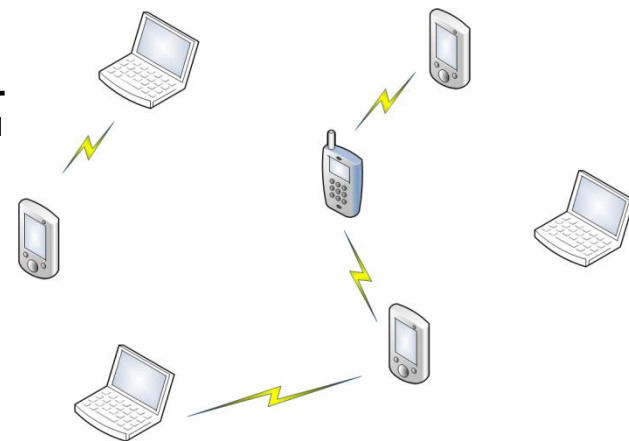
背景

▶ MANET(Mobile Ad-hoc Network)

- ノード同士がアドホックモードで接続しネットワークを構成
 - ノードはルータとして機能, 中継機能を持つ
 - 遠隔のノードとはマルチホップ通信
- 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

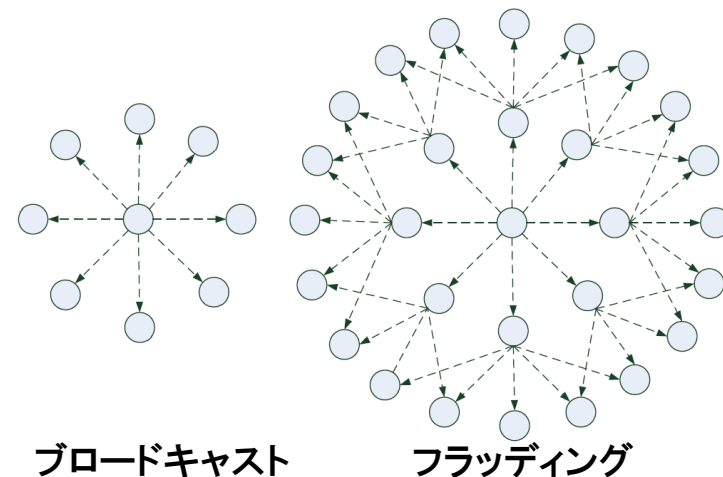


OLSRの概要

(Optimized Link State Routing)

- ▶ 通信要求発生前からRTを生成しておくProactive型のプロトコル
- ▶ 各ノードは定期的に制御メッセージを送受信し, 周辺ノードの情報を収集することによってRTを生成
- ▶ 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

制御メッセージにトラフィックの情報は含まれていない



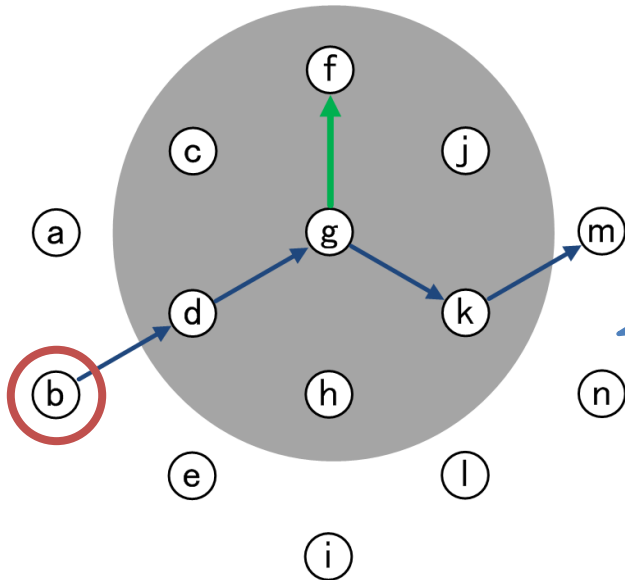
OLSRのRT

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

ノードbのRT

ノードbのRT



OLSRで生成される経路例

ノード数 : 14台
電波到達範囲: 1 hop先まで
既に行われている通信: $g \rightarrow f$

Dest	Next	hop
a	a	1

Dest	Next	hop
a	a	1

g	d	2
h	d	2
i	e	2
j	d	3

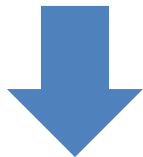
g	d	2
h	d	2
i	e	2
j	d	3

ホップ数を基準に経路が選択されるため、最短経路が複数存在する場合においても、高トラフィックのノードを経由する経路が生成される可能性がある

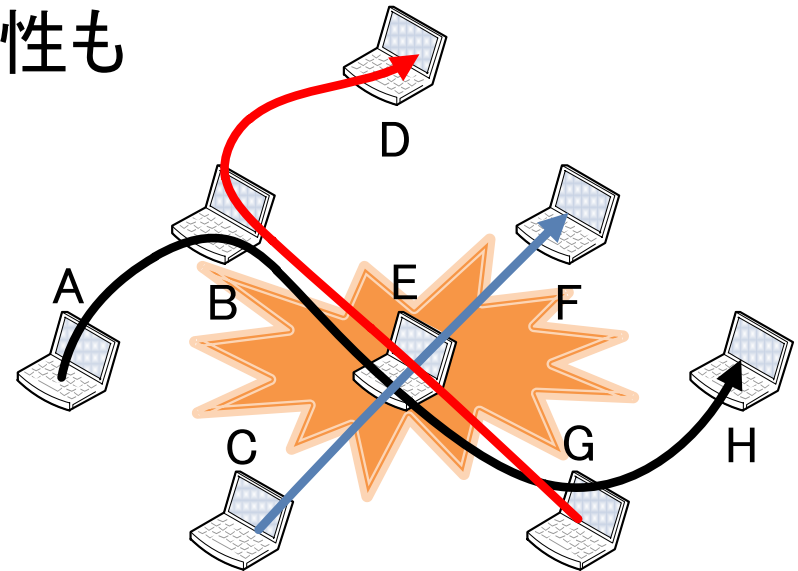
課題

- ▶ 多くのアドホックルーティングプロトコルは、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実装に任されている
- ▶ 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



提案方式:PD-OLSR

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

- ▶ OLSRを改造
- ▶ TCP用とUDP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可能

提案方式:PD-OLSR

(Proposed Protocol: PD-OLSR)

▶ 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

RTを分けることで性質を経路生成に反映

PD-OLSR流れ

トラフィックを計算

制御メッセージに載せて広告

ノードのトラフィック→リンクメトリックに変換

ダイクストラ法による経路探索

RMT(Route Metric Table)に探索結果を保存

RMTからRTを作成

今回は
UDPIについて説明

PD-OLSR流れ

トラフィックを計算

制御メッセージに載せて広告

ノードのトラフィック→リンクメトリックに変換

ダイクストラ法による経路探索

RMT(Route Metric Table)に探索結果を保存

RMTからRTを作成

PD-OLSR トラフィック情報

- ▶ トラフィックの高いノードを避けるための経路選択指標
- ▶ UDP: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:TCP Session
 - 自身が検出するTCPセッション数の合計

PD-OLSR流れ

トラフィックを計算

制御メッセージに載せて広告

ノードのトラフィック→リンクメトリックに変換

ダイクストラ法による経路探索

RMT(Route Metric Table)に探索結果を保存

RMTからRTを作成

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

PD-OLSR流れ

トラフィックを計算

制御メッセージに載せて広告

ノードのトラフィック→リンクメトリックに変換

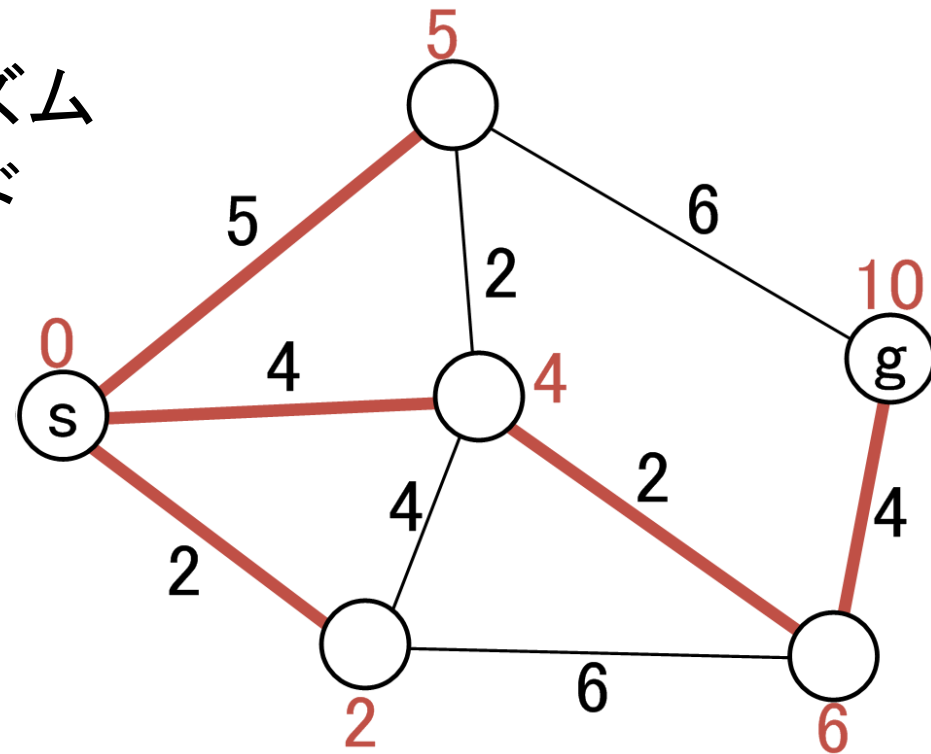
ダイクストラ法による経路探索

RMT(Route Metric Table)に探索結果を保存

RMTからRTを作成

ダイクストラ法(Dijkstra's Algorithm)

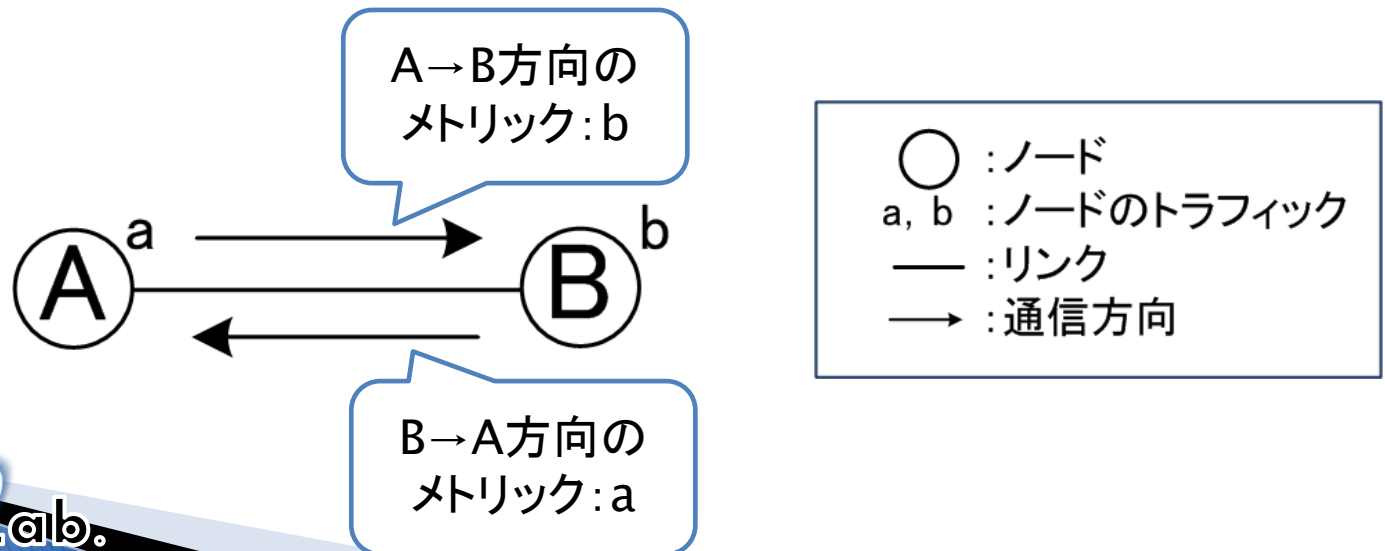
- ▶ 最短経路問題を効率的に解く
グラフ理論におけるアルゴリズム
- ▶ スタートノードからゴールノード
までの最短経路とその距離を
求めることができる



最短経路に限らずホップ数を増やした経路を選択可能に

双方向で異なるリンクメトリック

- ▶ ノードのトラフィックの情報からダイクストラ法で用いるリンクメトリックへ変換
 - 双方向で異なるメトリックに変換
 - リンクの宛先方向のノードのトラフィックをメトリックとする
 - 双方向で同じメトリック($a+b$ など)と比較して更新に必要な情報を少なくすることができる



PD-OLSR流れ

トラフィックを計算

制御メッセージに載せて広告

ノードのトラフィック→リンクメトリックに変換

ダイクストラ法による経路探索

RMT(Route Metric Table)に探索結果を保存

RMTからRTを作成

RMT(Route Metric Table)

- ▶ 新たに定義するテーブル
- ▶ ダイクストラ法による経路探索結果を保存
- ▶ 構成
 - Dest:宛先ノード
 - Cost:コスト(=経路の合計メトリック)
 - hop:ホップ数
 - hop1,hop2...:中継ノード

Dest	Cost	hop	hop1	hop2	...
a	0	1	a		
c	4	2	a	c	
d	4	1	d		
e	0	1	e		
⋮					

PD-OLSR流れ

トラフィックを計算

制御メッセージに載せて広告

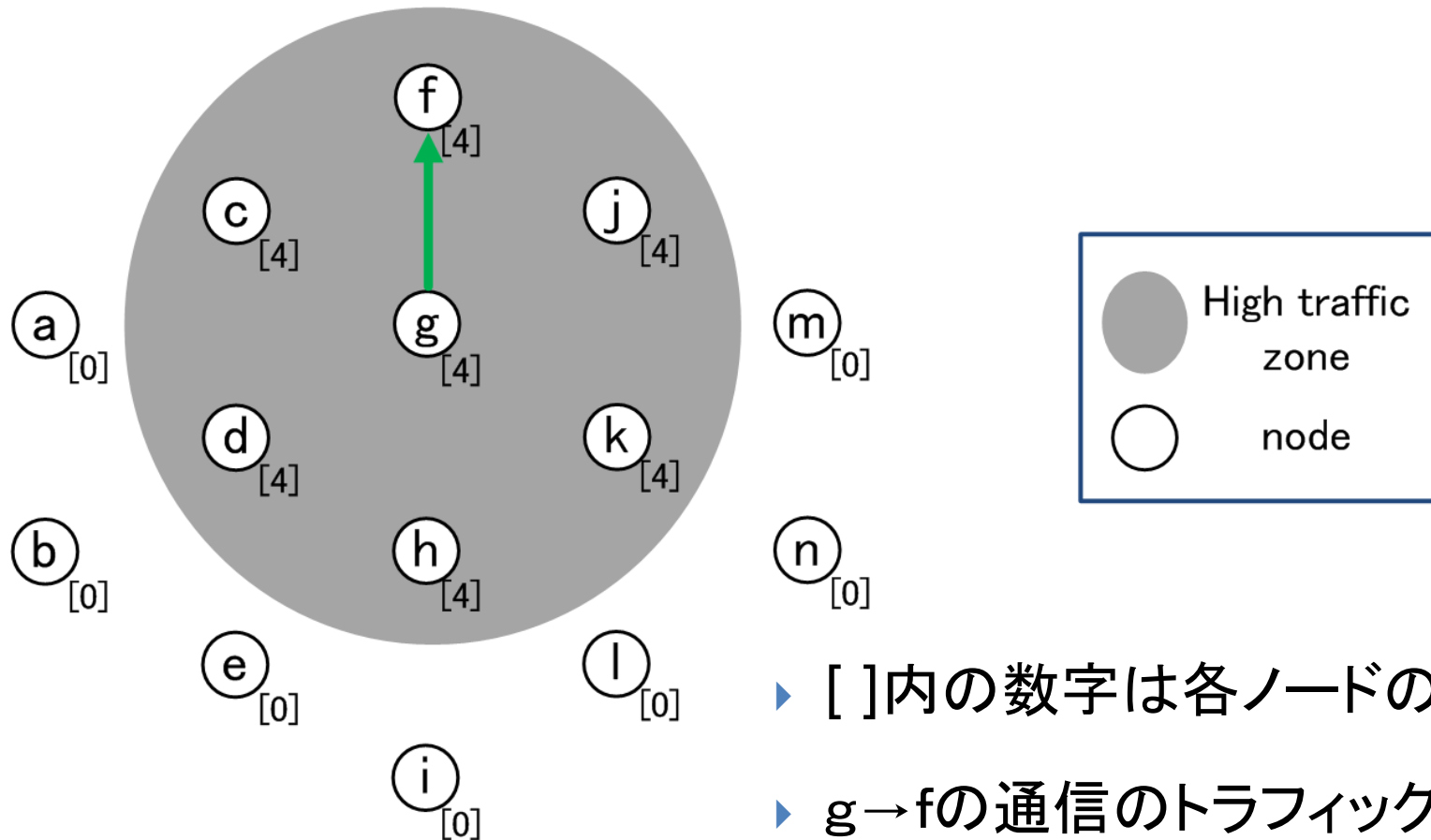
ノードのトラフィック→リンクメトリックに変換

ダイクストラ法による経路探索

RMT(Route Metric Table)に探索結果を保存

RMTからRTを作成

PD-OLSR 動作



- ▶ []内の数字は各ノードのトラフィック
- ▶ $g \rightarrow f$ の通信のトラフィック量: 4

PD-OLSR 動作

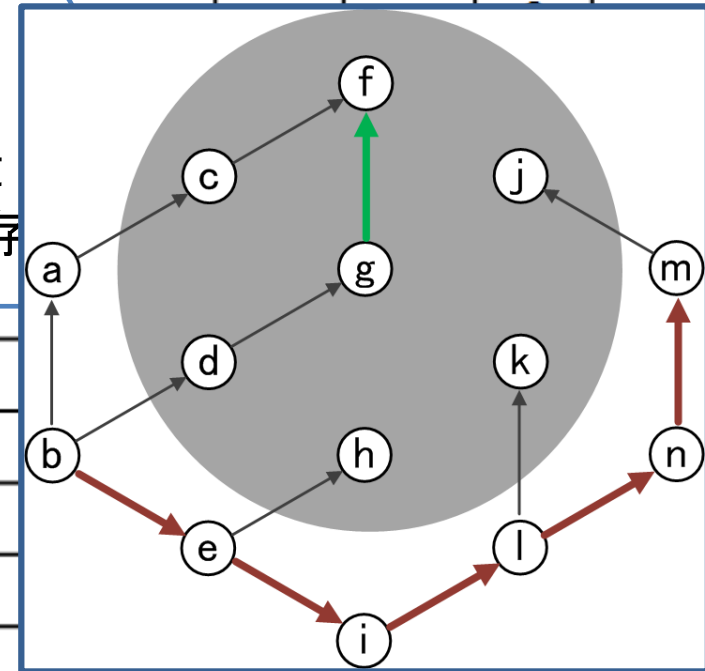
ノードbのRMT

Dest	Cost	hop	hop1	hop2	hop3	hop4	hop5	hop6
a	0	1	a					
c	4	2	a	c				
d	4	1	d					
e	0	1	e					
f	8	3	a	c	f			
g	8	2	d	g				
h	4	2	e	h				
i	0	2	e	i				
j	4	6	e	i	l	n	m	j
k	4	4	e	i	l	k		
l	0	3	e	i	l			
m	0	5	e	i	l	n	m	
n	0	4	e	i	l	n		

Dest→Dest
hop→hop
hop1→Next
としてRTに保存

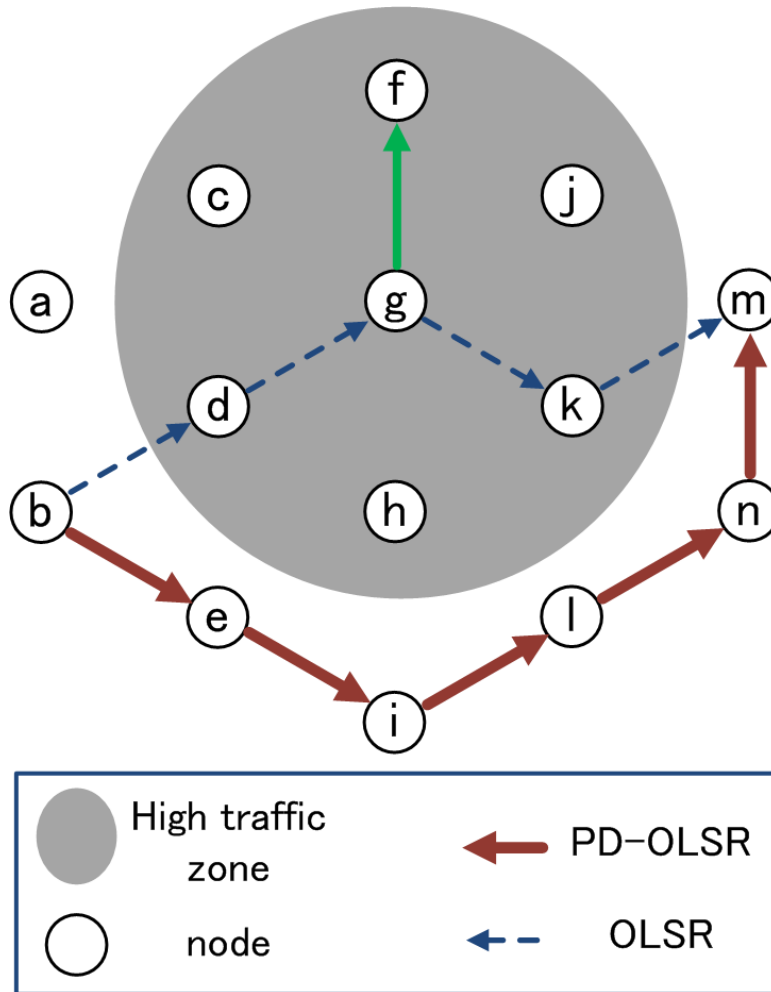
ノードbのRT

Dest	Next	hop
a	a	1



m	e	5
n	e	4

経路比較



- ▶ トラフィックの高いノードを避けた経路を生成可能
- ▶ 経路コストがより小さければホップ数を増やした経路を選択可能
 - 例での各経路のホップ数
 - OLSR : 4hop
 - PD-OLSR : 5hop

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
- ダイクストラ法を取り入れることで冗長経路の選択が可能となる

▶ 今後

- 提案方式をシミュレータに実装し, RTをUDPとTCPで分けたことによる効果を検証する
- 冗長経路についてホップ数増加をどの程度許すのかについて検討



補足資料

PD-OLSR

▶ 経路選択指標

- UDP: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:TCP Session
 - キャリアとして検出するTCPセッション数と実際に自身が処理しているTCPセッション数の合計

▶ UDP通信の経路

- 単純にUDP Trafficが最小の経路を選ぶ

▶ TCP通信の経路

- TCP の特性を活かして TCP スループットの公平性がとれる経路選ぶため, TCP Sessionが最小の経路を選ぶ

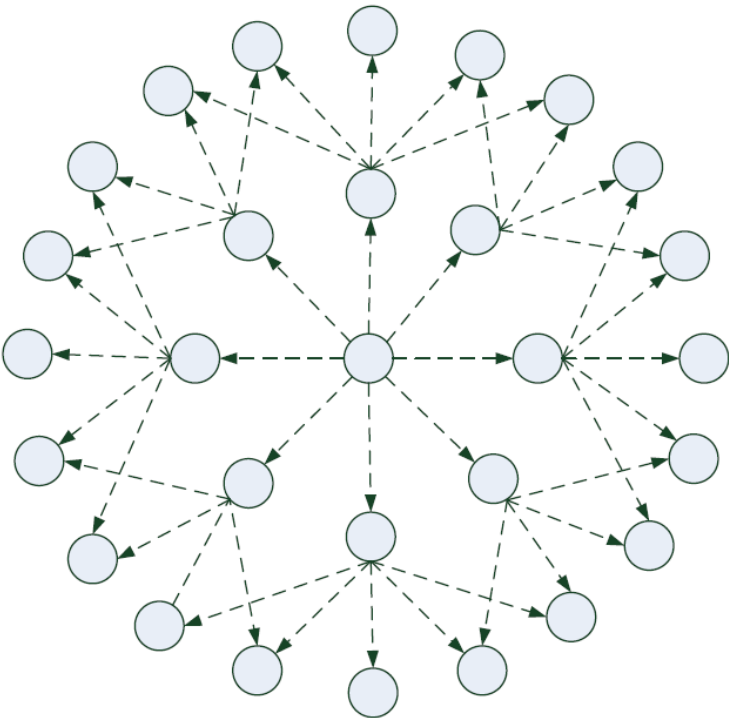
冗長経路についての検討内容

- ▶ ホップ数についてどのように上限を設けるのか
 - 固定値での上限
 - 経路のホップ数の一定割合
 - ネットワークの端末数

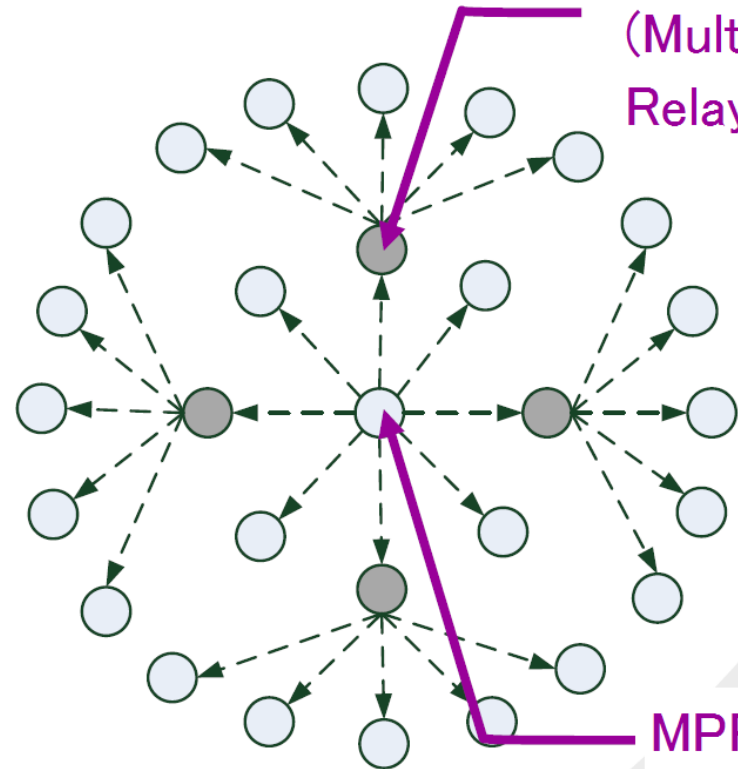
OLSRのフラッディング

MPR
(Multipoint
Relay)

MPRセレクト



通常のフラッディング



OLSRのフラッディング

アドホックルーティングプロトコル

▶ プロアクティブ型

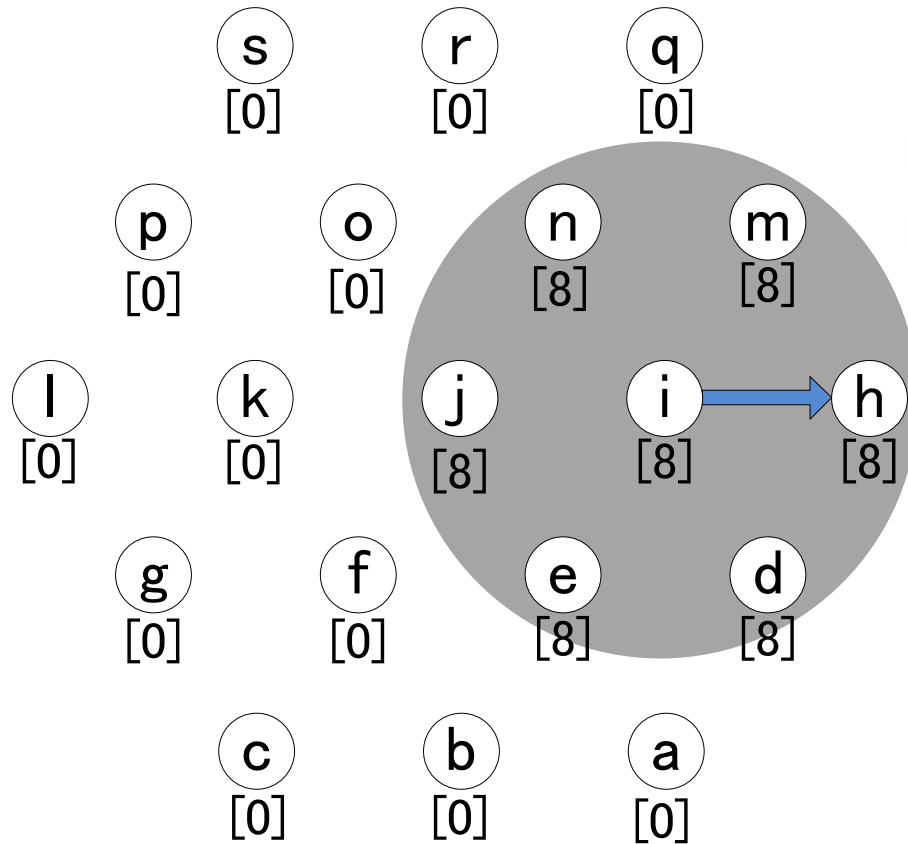
- 通信要求が発生する前からRTを生成
- ノードの移動が少なく, 通信頻度の高いネットワークに適する
- 例: OLSR(Optimized Link State Routing)

▶ リアクティブ型

- 通信要求が発生した際にネットワーク内で経路を探索する
- ノードの移動が頻繁なネットワークに適する
- 例: AODV(Ad hoc On-Demand Distance Vector)

トラフィックを考慮した 複数の最短経路からの経路選択

PD-OLSR 動作 1



- ▶ 各ノードがトラフィックを計算
- ▶ HELLOメッセージ, TCメッセージにトラフィック情報を追加し, ネットワーク全体に通知



[]内の数字は各ノードのトラフィック

PD-OLSR 動作 2

同一Destで複数の次ホップがある場合Trafficが最小となる次ホップを選択する

- ▶ RMT(Route Metric Table)を新たに定義
 - Dest, Next(Next node), hop数, Trafficから構成される
 - 1つのDestに対して全てのNextが保存される
- ▶ RMTからRTを生成

高トラフィックのノードを回避する経路の次ホップがRTに保存される

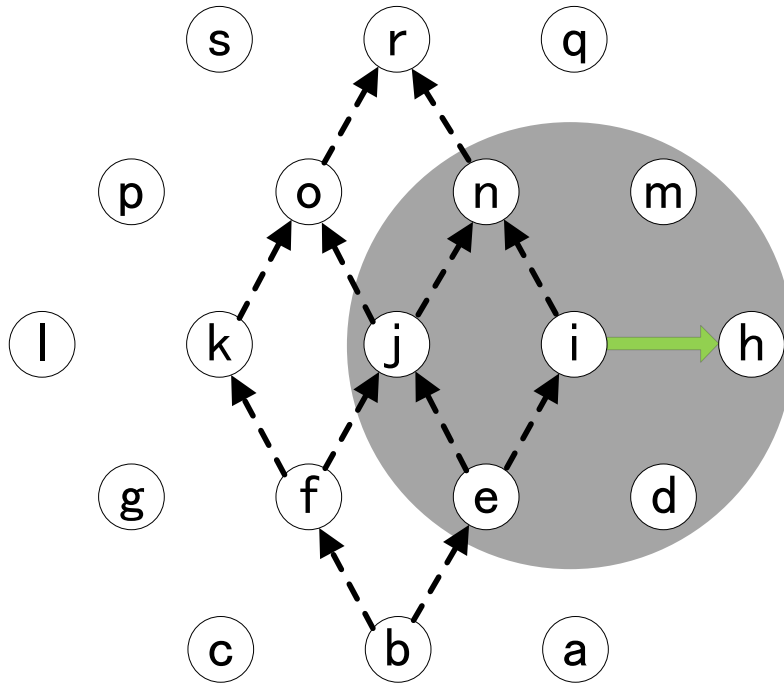
Dest	Next	hop	Traffic
a	a	1	0
c	c	1	0
d	a	2	0
d	e	2	8
e	e	1	8
f	f	1	0
省略			
q	e	4	8
q	f	4	0
r	e	4	8
r	f	4	0
s	e	4	8
s	f	4	0

ノードbのRMT

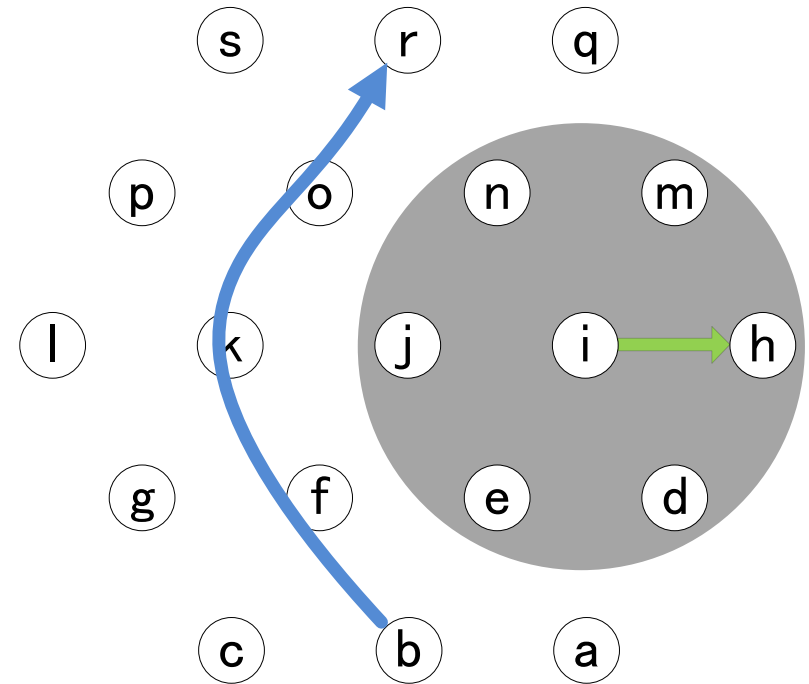
Dest	Next	hop
a	a	1
c	c	1
d	a	2
e	e	1
f	f	1
g	c	2
h	a	3
i	e	2
j	f	2
k	c	2
l	e	3
m	f	3
n	f	3
o	f	3
p	f	3
q	f	4
r	f	4
s	f	4

ノードbのRT

生成される経路の比較



OLSRによって生成される経路



PD-OLSRによって生成される経路

- ▶ トラフィックの高いノードを避けたルーティングを行うことができる

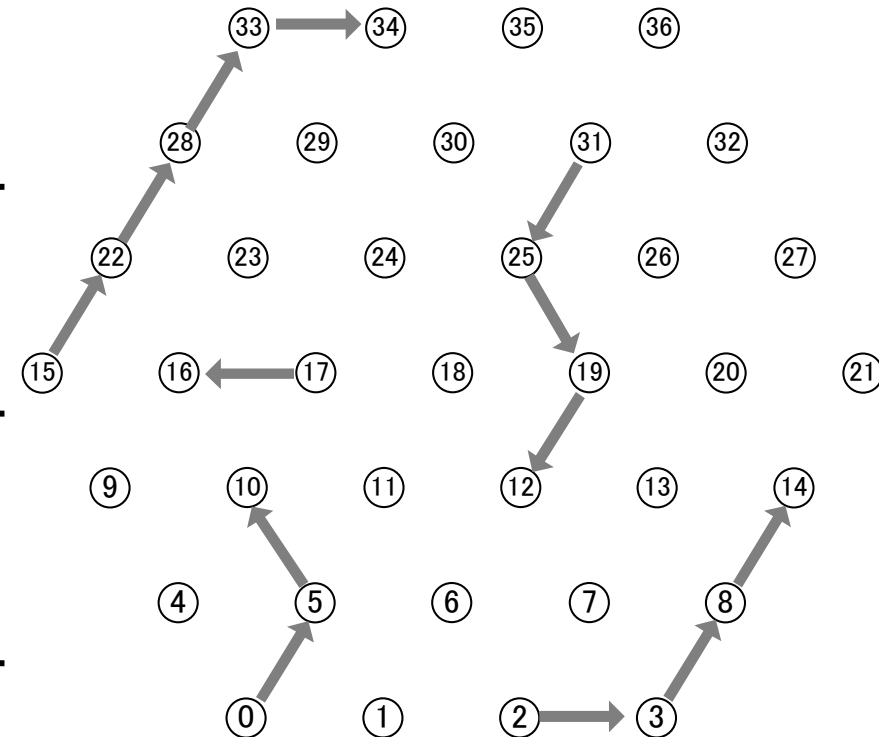
評価

- ▶ UDP用RT生成機能をネットワークシミュレータns-2に実装
- ▶ UDP通信のシミュレーション評価を行った

シミュレーション評価

環境

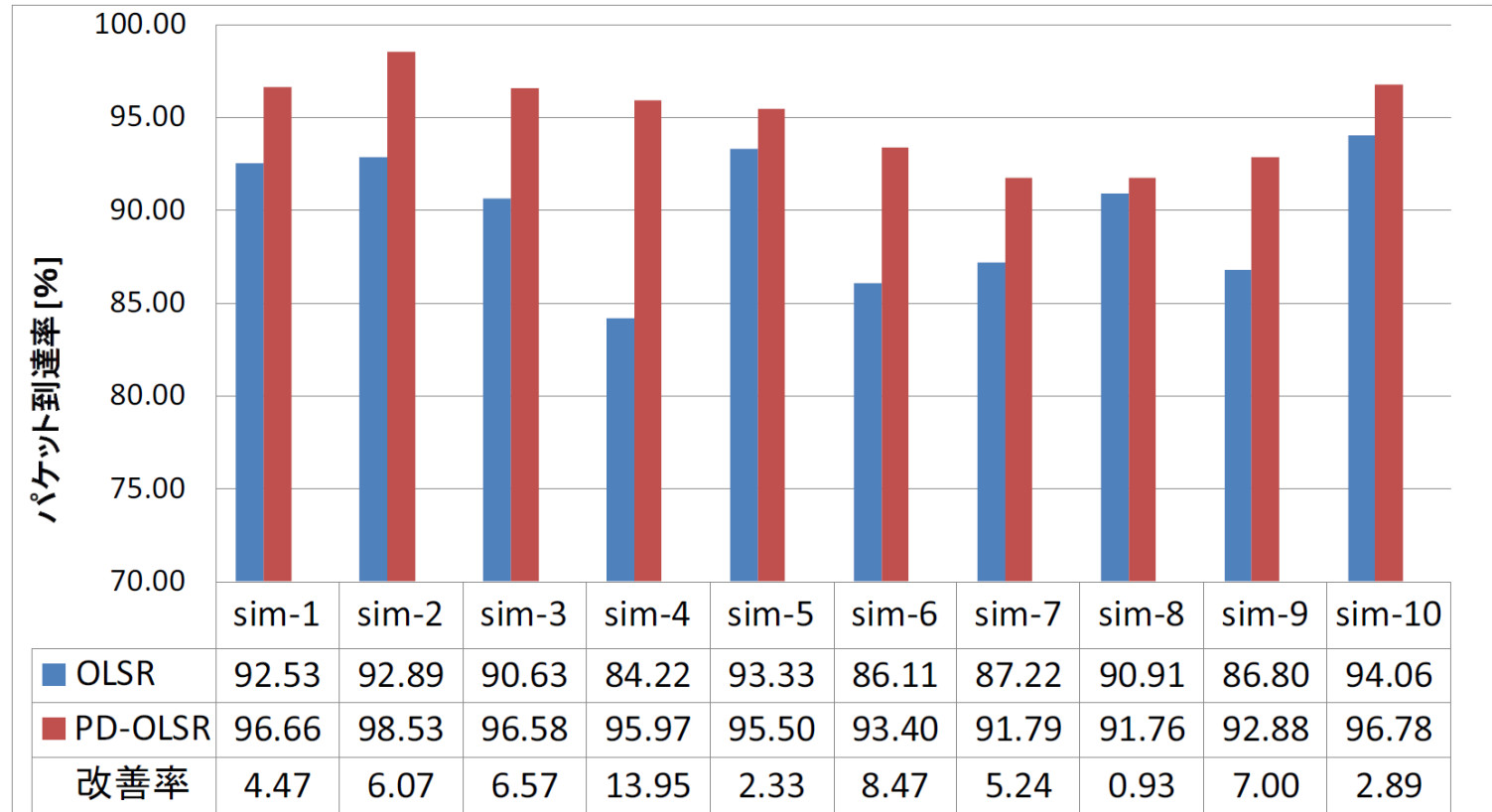
アドホック ネットワーク	ノード数 電波到達範囲 ノード間距離 ルーティングプロトコル 無線規格	37 [台] 100 [m] (1 hop) 95 [m] OLSR, PD-OLSR 802.11g
VoIPを想定した UDP通信	台数 選び方 通信タイプ トランスポートプロトコル パケットサイズ データ転送量	2台1ペア ランダム CBR UDP 200 [Byte] 64 [Kbps]



シミュレーション開始30秒後からUDPセッションを10秒間隔毎に増加させていく
合計530秒間のシミュレーションをOLSRを使った場合とPD-OLSRを使った場合
で10回ずつ行い、ネットワーク全体のパケット到達率を比較

シミュレーション評価

▶ 結果



- どのシミュレーションでもPD-OLSRによるパケット到達率の改善が見られた
- PD-OLSRの方がOLSRに比べ、ネットワーク全体のパケット到達が平均で約6%改善された