

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

森崎 明^{†1} 渡邊 晃^{†1}

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、経路生成の際に TCP や UDP の通信状況が考慮されていない。そのため、最短経路が複数存在する場合にはどの経路を選ぶかは実装に依存したものとなっている。本稿では OLSR (Optimized Link State Routing) を拡張することにより、TCP, UDP それぞれの特性を活せる経路選択が可能なアドホックルーティングプロトコルを提案する。

A proposal on an Ad-hoc Routing Protocol considering Traffic Condition

AKIRA MORISAKI^{†1} and AKIRA WATANABE^{†1}

With the spread of mobile nodes, studies on MANET (Mobile Ad-hoc Network) that can build networks solely with mobile nodes are drawing much attention. However, most of the ad-hoc routing protocols have not considered the traffic conditions in the network. Thus, we propose an ad-hoc routing protocol consider traffic conditions by way of extending OLSR (Optimized Link State Routing).

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。無線 LAN を構築する方法には、端末が必ず AP (AccessPoint) を介して通信を行うインフラストラクチャモードによる方法と、端末同士で直接通信を行うアドホックモードによる方法がある。後者は、災害時やイベント会場などで一時的な無線ネットワークを構築できるモバイルアドホックネットワーク (MANET: Mobile Ad-hoc Network)¹⁾ に応用されている。MANET は、あらゆる無線端末が中継端末となり得るため、その場でネットワークを構築することができるという特徴がある。近年では、インフラストラクチャーモードの AP 間を MANET の技術で結合する無線メッシュネットワークの研究にも注目が集まっている²⁾⁻⁶⁾。

MANET を構築するには、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) により、現在まで多くの方式が提案されているが⁷⁾⁻¹³⁾、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) を探索することが目的となっており、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してしまうという課題がある¹⁴⁾。

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)¹⁵⁾ の経路選択では、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されるため、安定した経路により通信を行うことができる。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)¹⁶⁾ は DSR (Dynamic Source Routing Protocol)⁸⁾ 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。しかし、この方式は 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。

^{†1} 名城大学大学院理工学研究科

Graduate School of Science and Technology, Meijo University

本稿では、MANET の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 を拡張することによって、経路上の通信状態を考慮したプロトコル PD-OLSR (Protocol Dependent-OLSR) を提案する。具体的には、TCP と UDP の特性を活かせるように、TCP と UDP 用のルーティングテーブルを別々に生成する。UDP の場合はトラフィックを避けた経路、TCP の場合は TCP スループットの公平性がとれた経路で通信を行うことができる。経路生成を実現する。今回は UDP に係わる部分についてネットワークシミュレータ ns-2 によるシミュレーションを行い、その効果を確認した。

以下、2章では MANET のルーティングプロトコルの分類と OLSR の概要について説明する。3章では PD-OLSR の経路生成方法、4章で PD-OLSR のシミュレータへの実装方法、5章でシミュレーションによる PD-OLSR の評価を示し、最後に6章でまとめを行う。

2. 既存のアドホックルーティングプロトコル

2.1 アドホックルーティングプロトコルの分類

MANET では電波到達範囲外の移動可能ノードと通信するため、各ノードは中継機能を持ち、ノードの移動によるリンク接続状態の変化に迅速に対応する必要がある。MANET には様々な用途が考えられ、用途に応じたルーティングプロトコルが存在する。これまで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、全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、表 1 に示すように 3 種類に分類することができる。これらは、その特徴を活かせる環境によって使い分けられる¹⁷⁾⁻²⁰⁾。

2.2 OLSR

プロアクティブ型のプロトコルは、ルーティングテーブル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、シンプルに経路上の通信状況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でかつ最も普及している OLSR (Optimized Link State Routing)⁷⁾ を提案方式のベースとする。以下に OLSR の原理と経路生成方法について説明する。

2.2.1 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) に隣接ノードにブロードキャストする。HELLO メッセージは自身のアドレス、シーケンス番号、隣接ノードのアドレスなどの情報をもっている。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及び隣接ノードの更に隣接ノード、すなわち 2 ホップ先のノード (以

表 1 MANET のルーティングプロトコルの分類
 Table 1 The classification of the routing protocol of MANET

分類	方式	特徴	例
Proactive 型	通信要求が発生する前からルーティングテーブルを生成	通信頻度の高いネットワークに適する	OLSR DSRV TBRPF
Reactive 型	通信要求が発生した際にネットワーク内で経路探索プロセスが始動	ノードの移動が頻繁なネットワークに適する	AODV DSR TORA ABR
Hybrid 型	ネットワーク内を複数のゾーンに分割し、ゾーンの内外で Proactive 型と Reactive 型を使い分けて経路を構築	Proactive 型と Reactive 型の両方の特徴を活かすことができる	ZRP

後、2 ホップ隣接ノード) のアドレスを得ることができる。

2.2.2 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を実現できることである。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。OLSR では必要最低限の中継ノード (MPR : Multipoint Relay) を選択し、この中で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 メッセージで隣接ノードに通知する。

2.2.3 トポロジー情報の配送

OLSR はトポロジー情報を定期的に TC (Topology Control) メッセージによって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 メッセージの送信間隔はデフォルト値で 5 秒である。

2.2.4 各ノードが持つ情報

各ノードはルーティングテーブル (以後、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、トポロジー集合などの 7 つのテーブルからなるリポジトリを持つ。これらのテーブルは隣接ノードだけに届く HELLO メッセージ、ネットワーク全体にフラッディングされる TC メッセージによって生成される。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、双方向リンクの 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。トポロジー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。

2.2.5 経路計算

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 OLSR の経路生成手順を示す。簡単のためノードは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 1 において、RT はノード b が持つ RT であり、左側の RT はノード a からノード s のうち、ノード a からノード q までの経路が途中まで作成された状態、右側の RT はさらにノード r とノード 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、Dest が r となるレコードの Next には r の隣接ノードであるノード n とノード o のうち、右側の RT のように RT を上から順に探索したときに、最初に発見されるノード n の Next であるノード e が設定される。Dest がノード s の経路も同様に追加される。

同様の方法で全ノードの RT が生成されると、ノード r への経路が決まり、図 1 右に示す青経路 [b → e → i → n → r] という 1 つの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。すなわち、選択される経路は実装に依存したものとなっている。

もし、ノード i からノード h への通信が既に行われている状態で、ノード b からノード r への通信が青経路で行われると、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、新たなトラヒ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。

3. PD-OLSR

3.1 PD-OLSR の概要

本稿では OLSR をベースにし、通信状態を考慮してプロトコル毎に効率の良い経路選択を行う 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用い、経路選択指標については、TCP 通信と UDP 通信の特性の違いに着

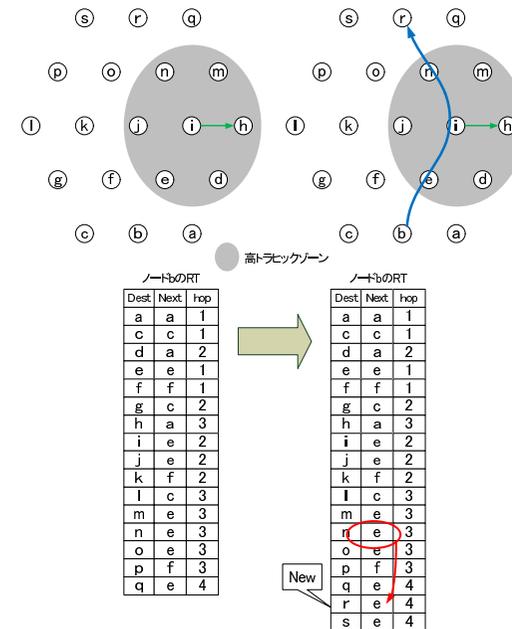


図 1 OLSR による RT 生成手順
 Fig. 1 RT generation procedure by OLSR

目した。

TCP/IP では UDP と TCP という特性が全く異なる 2 種類のラヒックが存在する。UDP は端末側が意図した流量のトラヒ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てもそれは変わらない。これに対して、TCP は輻輳制御の機能により、順調に ACK が返ってくれば、ウイ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ものと判断し、ウインドウサイズを小さくする。このようにしてウインドウサイズが適切な大きさに調整され、ネットワークがさらに輻輳することを防止する。このような特性の違いから、ネットワーク上のトラヒックは、以下ようになる。まず送信された UDP パケットの合計より UDP が占めるトラヒック量が定まり、残りの余裕がある帯域分を複数の TCP セッションが分け合う。UDP のパケットロスはそのまま消滅するが、TCP の場合は再送処理を行いながら、スループットが最大になるように輻輳制御が働く。TCP の効率性は、TCP の輻輳制御がうま

く機能するかどうかによって決まる²¹⁾⁻²³⁾.

以上の考えに基づき、UDP 通信と TCP 通信の経路選択指標を別々に考える。UDP 通信においては、単純に現在の UDP トラフィックの少ない経路を選ぶ。一方 TCP 通信においては、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を選択する。そのため、TCP セッション数の少ない経路を選ぶ。

3.2 PD-OLSR の経路計算

PD-OLSR では UDP 通信用と TCP 通信用の RT を別々に生成する。それぞれの RT の経路選択指標は UDP Traffic と TCP Session とする。UDP Traffic とは自身が検出するネットワーク上のキャリアの総量であり、TCP Session とは自身が検出する TCP セッション数の合計である。以下、図 2 を用いて 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。

図 2 のアドホックネットワークの条件は図 1 の場合と同じである。図 2 左のテーブルは各ノードが計算した UDP Traffic の情報である。ノード i からノード h への UDP 通信が行われているためノード d, e, h, i, j, m, n では 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 8 としている。図 2 中央のテーブルは UDP 通信用の RT を生成するためノード b が持つ、新たに定義した経路計算テーブル (RCT:Route Calculation Table) である。RCT は宛先ノード (Dest), 宛先への次ホップノード (Next), ホップ数 (hop), Next の UDP Traffic から構成され、最短経路候補を複数有している。図 2 右のテーブルはノード b が持つ、RCT をもとに生成された UDP 通信用 RT である。

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 メッセージに乗せて隣接ノードへ広告する。各ノードは HELLO メッセージと TC メッセージを受信すると、その情報を情報リポジトリの中のリンク集合、2 ホップ隣接ノード集合及びトポロジー集合に格納する。そして、情報リポジトリに格納された情報をもとに RCT を生成する。

RCT が生成されると、RCT の各 Dest に対して最小 UDPTraffic となる経路が UDP 用の RT へ設定される。同様に、各ノードで RCT から RT が生成されると各宛先への経路が完成する。例えば、図 2 でノード b からノード r へ UDP 通信が行われると高トラフィックゾーンを避けた青経路 [b → f → k → o → r] が生成される。TCP 通信用の RT 生成については、図 2 と上記の説明において、UDP Traffic を TCP Session に置き換えることで生成できるため省略する。

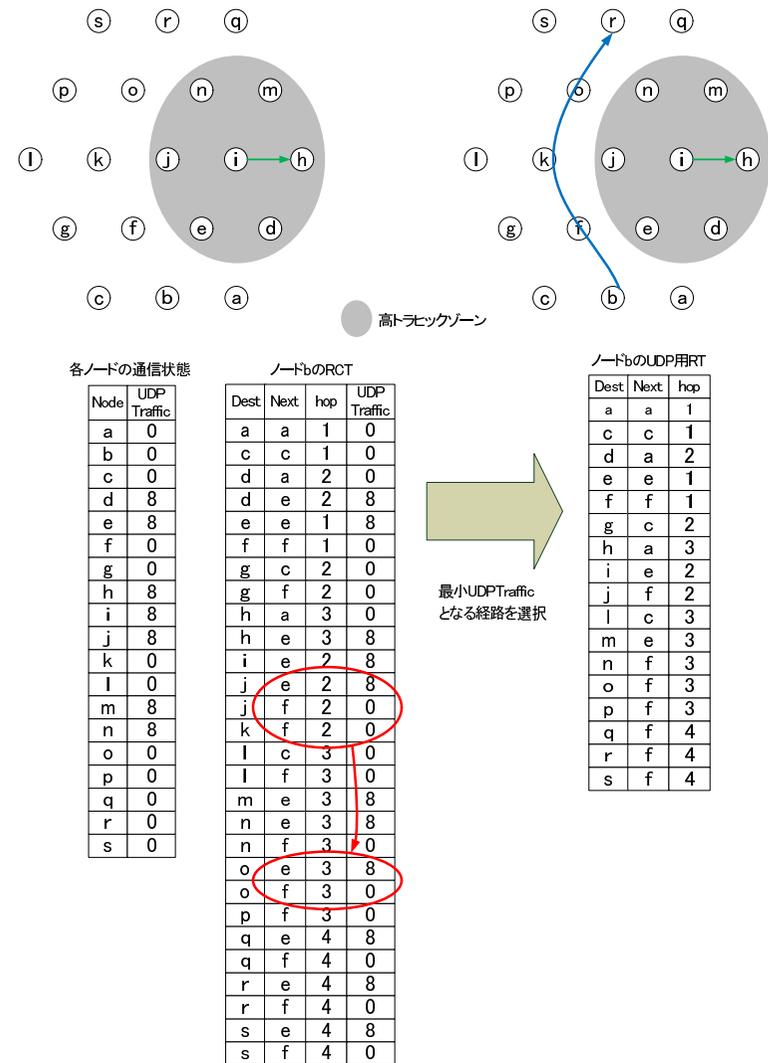


図 2 PD-OLSR による UDP 用 RT の生成手順
Fig. 2 Generation procedure of RT for UDP by PD-OLSR

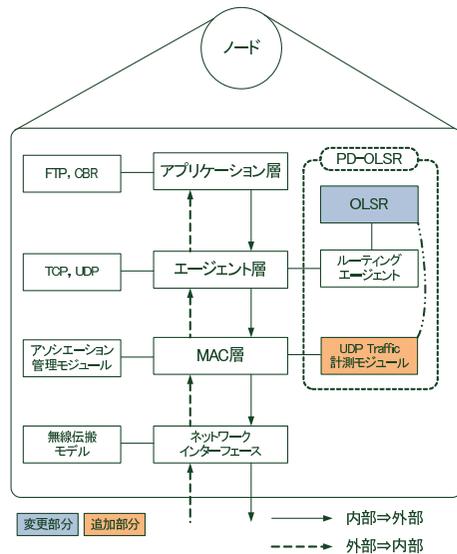


図 3 ns-2 の内部構造と変更部分
Fig. 3 Internal structure and a change part of ns-2

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-2²⁴⁾ へ実装する方法を検討した。今回は UDP 通信用の RT の生成を実装したので、その概要を以下に示す。

4.1 ns-2 の変更部分

図 3 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に拡張した。

4.2 OLSR 拡張方法

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は図 4 のようになっている。PD-OLSR の処理が行えるように、この処理に以下の拡張を行った。

- 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

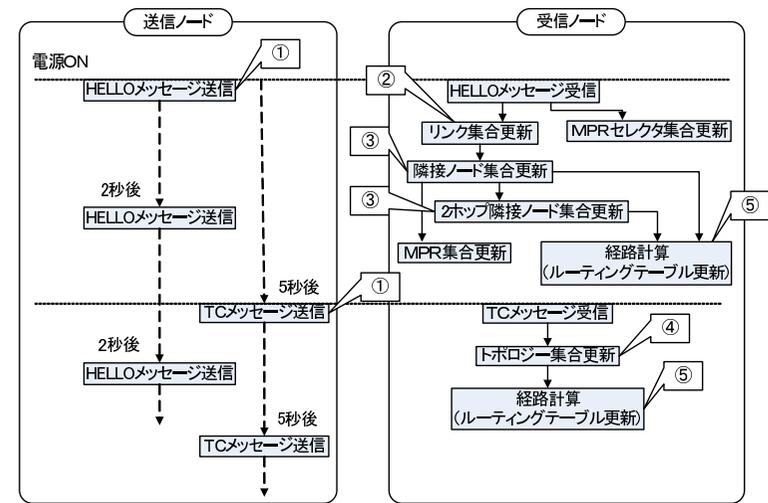


図 4 OLSR の拡張
Fig. 4 Extension of OLSR

- 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作成
- 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 トポロジー集合の更新
 - 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記録を作成
- ⑤ 経路計算
 - 経路計算 (RT 更新) プロセスに先立ち、3.2 節の方法で RCT を生成
 - RCT の中から UDP 通信用 RT を生成

5. 評価

PD-OLSRにおけるUDP通信RTの有用性を示すためのシミュレーションを行った結果を以下に示す。

5.1 動作検証

PD-OLSRのUDP通信のRT生成機能の動作検証を行った。

図1の状況を想定し、アドホックネットワークを表2のように構成した。図1においてUDP通信が2セッション行われるだけでは明確な違いが得られないため、ノードiからノードhへの通信をUDP、ノードbからノードrへの通信をTCPとし、TCPスループットの違いを比較した。それぞれの通信を表3のように設定した。シミュレーションの開始から終了までの時間を60秒とし、シミュレーション開始30秒後に背景負荷通信としてノードiからノードhへUDP通信を開始させ、シミュレーション開始45秒後にノードbからノードrへTCP通信を開始させた。以上のシミュレーション内容をOLSRを使用した場合とPD-OLSRを使用した場合において行った。

OLSRを使用した場合のノードbからノードr間通信の経路は[b → f → j → n → r]となり、背景負荷通信からのトラヒックを受けるノードjとノードnが経路に選択されていた。一方、PD-OLSRを使用した場合のノードbからノードr間通信の経路は[b → f → k → o → r]となり、背景負荷通信からのトラヒックを受けないノードのみで経路が生成されていた。図5にOLSRとPD-OLSRのノードbからノードr間のTCPスループットの変化を示す。TCPスループットの平均は、OLSRでは2.5Mbps、PD-OLSRでは3.9Mbpsであり、約1.5倍の違いがあった。以上より、PD-OLSRのUDP通信のRT生成機能が正しく実装されていることが分かった。

表2 シミュレーションパラメータ 1
 Table 2 Simulation parameter 1

アドホックネットワーク	
ノード数	19 [台]
電波到達範囲	100 [m]
端末間距離	95 [m]
フィールド	700 × 700 [m]
ルーティングプロトコル	OLSR, PD-OLSR
無線規格	802.11g

表3 シミュレーションパラメータ 3
 Table 3 Simulation parameter 3

ノード i → ノード h	通信タイプ トランスポートプロトコル パケットサイズ パケット発生率	CBR UDP 200 [Byte] 1 [Mbps]
ノード b → ノード r	通信タイプ トランスポートプロトコル パケットサイズ 最大衝突ウィンドウサイズ	FTP TCP 1000 [Byte] 20 [pkt]

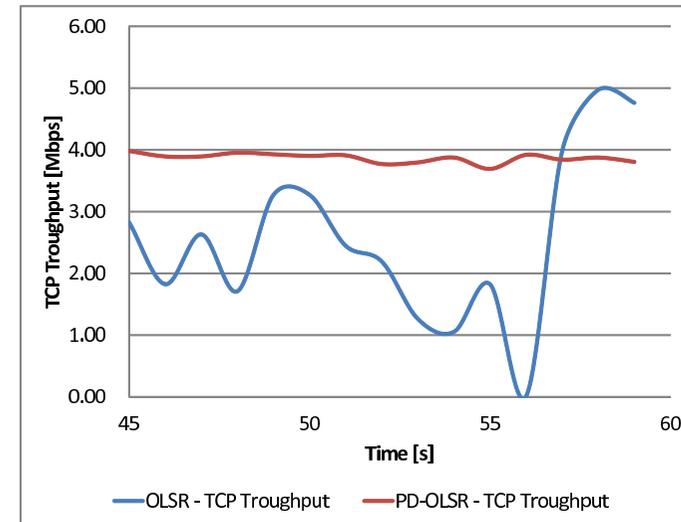


図5 ノードbからノードrのプロトコル毎TCPスループットの変化
 Fig. 5 TCP Throughput from node b to node r with each protocols

5.2 大規模シミュレーション

図 1 の構成を拡大して、ノードの数を 37 台とし、大規模なシミュレーションを試みた。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 がパケット到達率に与える影響を調べた。

5.2.1 シミュレーション条件

ノードの構成は表 2 と同様とした。UDP トラフィックの条件は表 4 の通りとした。シミュレーションの開始から終了までの時間を 140 秒とし、シミュレーション開始 30 秒後から 2 秒間隔で UDP セッションをネットワークが飽和し始めるまで増加させていった。UDP 通信を行うノードの組合せはランダムに選定した。以上のシミュレーション内容を合計 10 回を行い、その平均を求めた。

表 4 シミュレーションパラメータ 4
 Table 4 Simulation parameter 4

通信ノード	
台数	2 台 1 ペア
選び方	ランダム
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]

5.2.2 結果

ネットワーク全体で UDP セッションの送信元ノードが送信したパケット数の合計と宛先ノードが受信するパケット数の合計からネットワーク全体のパケット到達率を求め、OLSR と PD-OLSR を比較した。OLSR を用いた場合は UDP47 セッション程度でネットワークが飽和し始めるため、評価対象範囲を 47 セッションまでとした。図 6 に OLSR と PD-OLSR によるシミュレーション 10 回分の結果を示す。シミュレーション毎の改善率を求めた結果を表 5 に示す。PD-OLSR を用いると OLSR に比べ、約 8 % の改善が得られた。

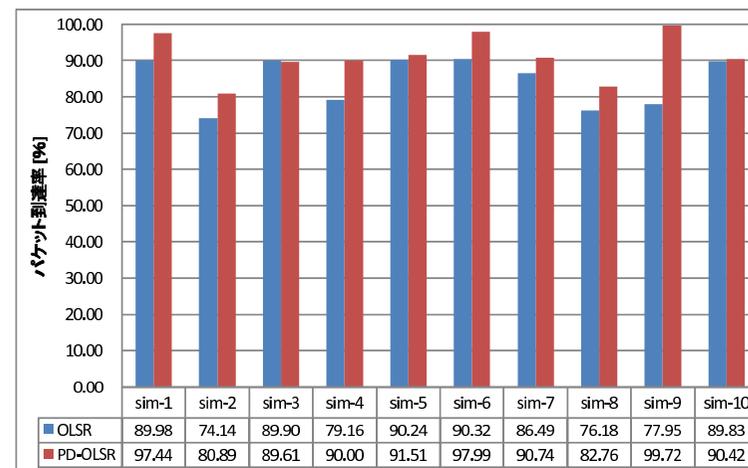


図 6 シミュレーション毎のパケット到達率
 Fig. 6 Packet Reachability at each simulation

表 5 シミュレーション毎の PD-OLSR による改善率
 Table 5 Improvement rate by PD-OLSR at each simulation

	改善 [%]
sim-1	8.29
sim-2	9.10
sim-3	-0.32
sim-4	13.70
sim-5	1.41
sim-6	8.49
sim-7	4.92
sim-8	8.64
sim-9	27.93
sim-10	0.66
Average	8.28

6. ま と め

OLSR を拡張することによって, 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. UDP 通信用の RT 生成機能をシミュレータに実装し, VoIP 通信を想定したシミュレーションを行った. その結果 UDP 通信においては, トラフィックの高い経路を避けた通信を行うことにより, パケット到達率が 8%程度向上することが分かった.

今後は, TCP 用の RT 生成機能をシミュレータに実装し, 動作検証を行う予定である. TCP ではスループットの違いとして表せるので, より明確な差が出ると期待している. また, 他のプロトコルに提案方式の機能を実装した場合や, 新たな経路選択指標と組合わせて経路生成が行える方法を検討する.

参 考 文 献

- 1) S. Corson: Mobile Ad hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, IETF (1999).
- 2) MetroMesh: <http://www.tropos.com/>.
- 3) MeshCruzer: <http://www.thinktube.com/>.
- 4) PacketHop: <http://www.packethop.com/>.
- 5) Y.Amir, C.Danilov, M.Hilsdale: Fast Handoff for Seamless Wireless Mesh Networks, *ACM MobiSys* (2006).
- 6) V.Navda, A.Kashyap, S.R.Das: Design and Evaluation of iMesh: An Infrastructure-Mode Wireless Mesh Network, *World of Wireless Mobile and Multimedia Networks*, pp.164–170 (2005).
- 7) T. Clausen, Ed: Optimized Link State Routing Protocol (OLSR), RFC 3626, IETF (2003).
- 8) D. Johnson: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728, IETF (2007).
- 9) C. Perkins: Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, IETF (2003).
- 10) R. Ogier: Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), RFC 3684, IETF (2004).
- 11) 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.
- 12) Charles E. Perkins, Pravin Bhagwat: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *ACM SIGCOMM*, Vol.24, No.4 (1994).
- 13) V.Park, S.Corson: Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification, Internet draft, IETF MANET Working Group (2001).
- 14) Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, Robert Morris: Performance of Multihop Wireless Networks: Shortest Path is Not Enough, *ACM SIGCOMM Computer Communication Review*, pp.83–88 (2003).
- 15) Toh, C.-K.: Associativity-Based Routing for Ad-Hoc Mobile Networks, *Wireless Personal Communications*, Vol.4, No.2, pp.103–139 (1997).
- 16) 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).
- 17) Royer, E.M., Chai-Keong Toh: A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks, *IEEE Personal Communications*, pp.46–55 (1999).
- 18) Daniel Lang: A comprehensive overview about selected Ad Hoc Networking Routing Protocols, Technical report, TUM, Department of Computer Science (2003).
- 19) 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).
- 20) 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-, 共立出版 (2003).
- 21) 伊藤将志, 鹿間敏弘, 渡邊 晃: 無線メッシュネットワークにおけるゲートウェイ分散方式の提案と評価, *DICOMO2008*, Vol.DICOMO2008, No.1, pp.1873–1879 (2008).
- 22) Masashi Ito, Toshihiro Shikama, Akira Watanabe: A Proposal of Gateway Decentralization Method in Wireless Mesh Networks and Its Evaluation, *ISITA2008* (2008).
- 23) Masashi Ito, Toshihiro Shikama, Akira Watanabe: Proposal and Evaluation of Multiple Gateways Distribution Method for Wireless Mesh Network, *ICUIMC* (2009).
- 24) The Network Simulator - ns-2: <http://www.isi.edu/nsnam/ns/>.

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

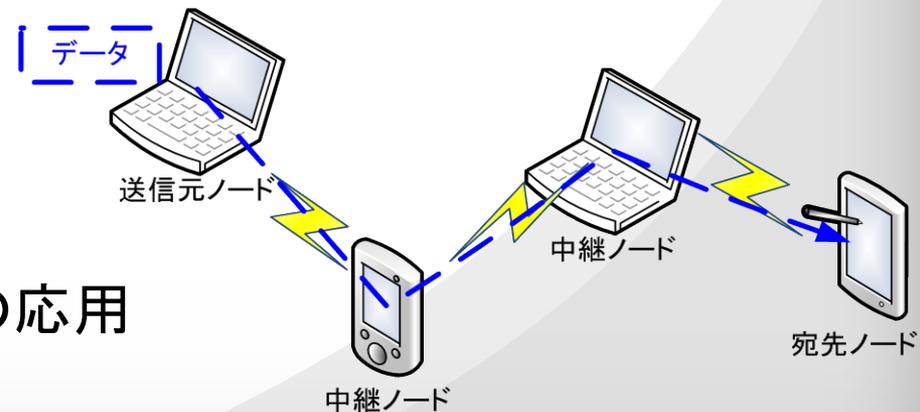
A proposal on an Ad-hoc Routing Protocol
considering Traffic Conditions

名城大学
森崎 明, 渡邊 晃

研究背景

MANET

- 無線LANの普及に伴い, MANET (Mobile Ad-hoc Network) の研究が期待されている
- MANET
 - アクセスポイントを必要としない
 - 無線通信機能を備えたノードのみで構成されるネットワーク
 - 無線通信に特化したアドホックルーティングプロトコルによって
 - ノードは中継機能をもつ
 - 遠隔のノードとはマルチホップ通信を行う
- 利用形態
 - インフラを利用できない環境
 - 災害時, イベント会場などの一時的な通信
 - 無線メッシュネットワークへの応用



アドホックルーティングプロトコル

分類	特徴
プロアクティブ型	<ul style="list-style-type: none">・通信要求が発生する前からルーティングテーブルを生成・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	<ul style="list-style-type: none">・通信要求が発生した際にネットワーク内で経路探索プロセスが始動・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)

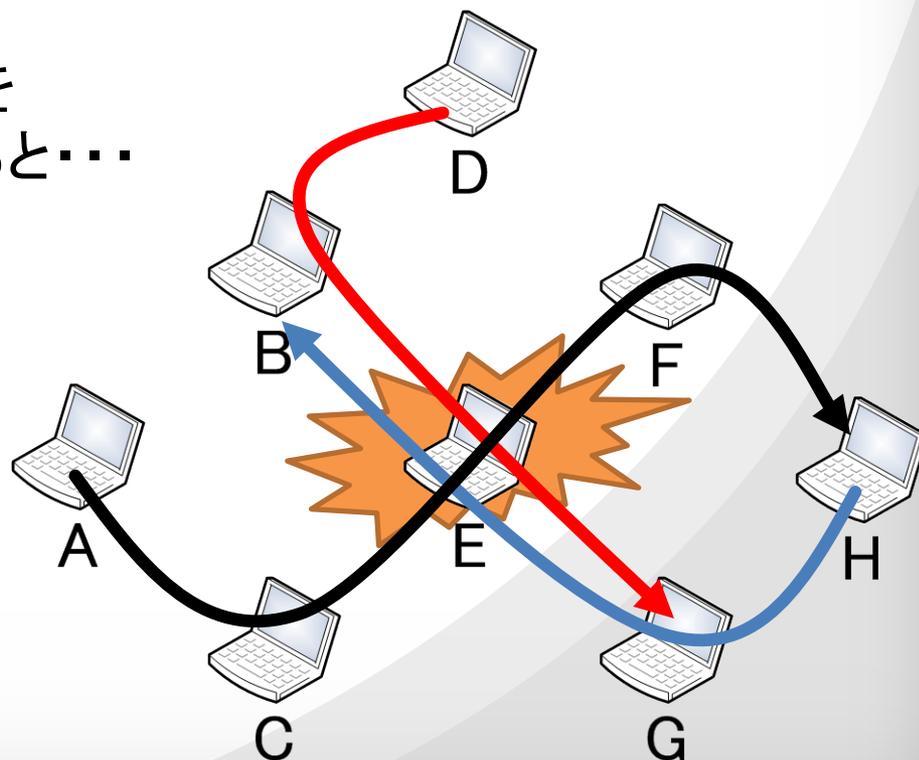
アドホックルーティングプロトコルの課題

- 多くのアドホックルーティングプロトコルは、中継ホップ数が最小となる最短経路を選択する
- 最短経路が複数存在する場合はどの経路を選択するかは実装に依存している
 - トラフィックが集中するノードを通る経路で通信が行われると…

パケットロスが多発



スループットが低下

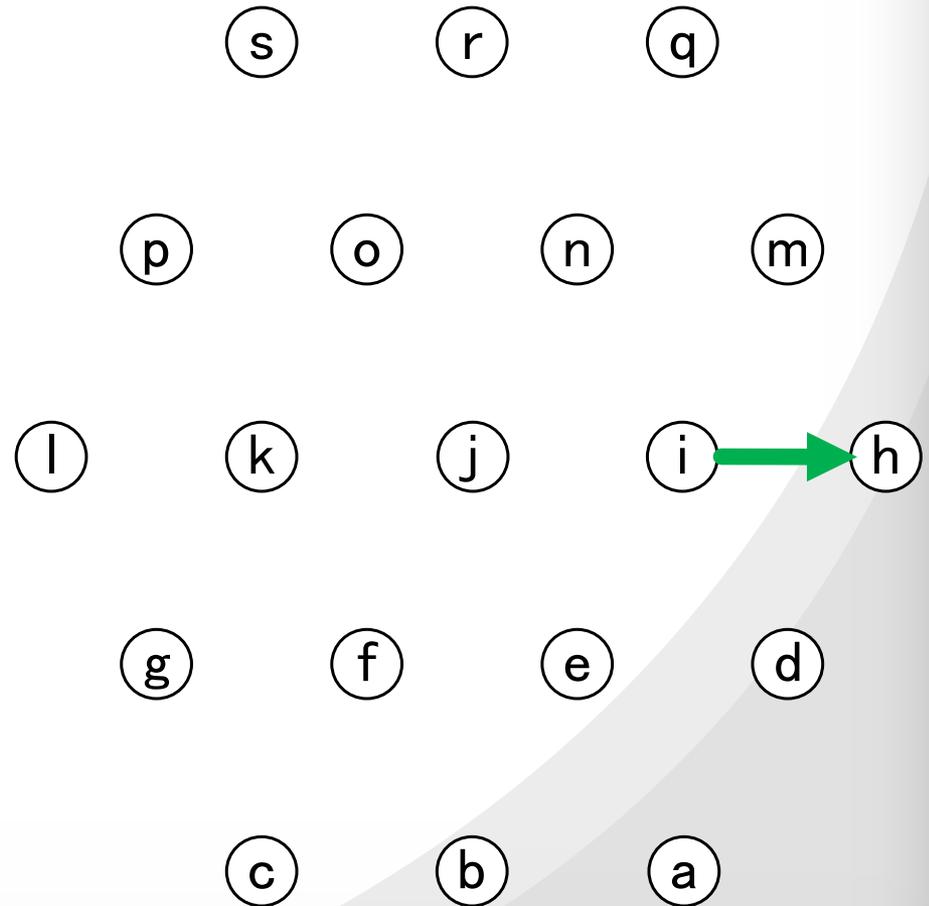


OLSR

OLSRの概要

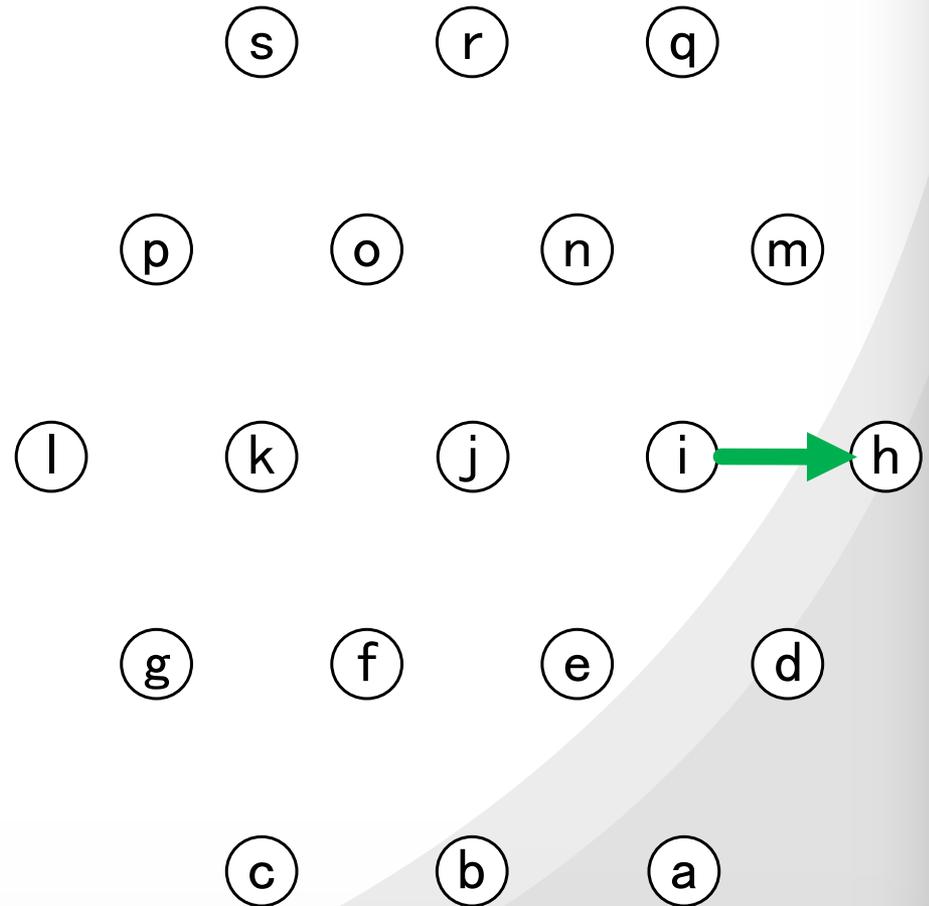
- 各ノードは定期的に制御メッセージを送受信し、周辺ノードの情報を収集することによって RT (Routing Table) を生成
- 制御メッセージ
 - HELLOメッセージ
 - 各ノードが持つ情報を通知するために、2秒毎に隣接ノードへブロードキャスト
 - TC (Topology Control) メッセージ
 - ネットワークトポロジーを通知するために、5秒毎にネットワーク全体にフラッディング

OLSRの経路生成



ノード数: 19台
電波到達範囲: 1ホップ先まで
既に行われている通信: $i \rightarrow h$

OLSRの経路生成

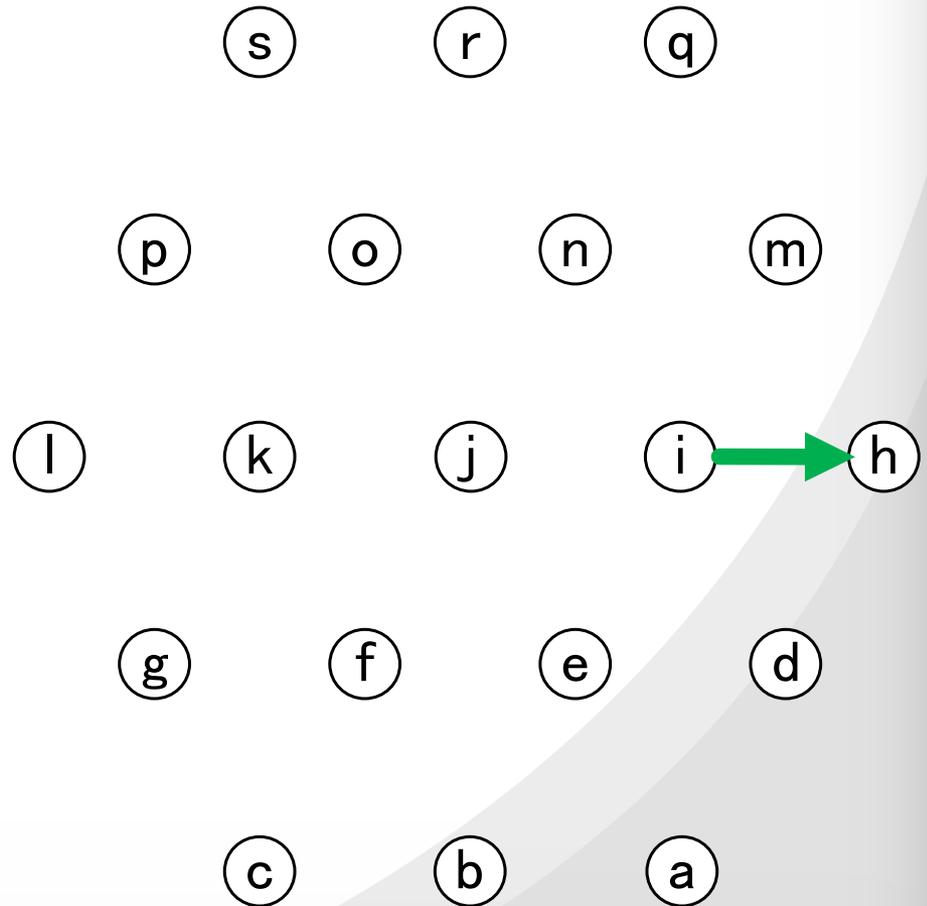


各ノードでHELLO, TCの送受信が行われ、ノードbのRTが生成

OLSRの経路生成

ノードbのRT

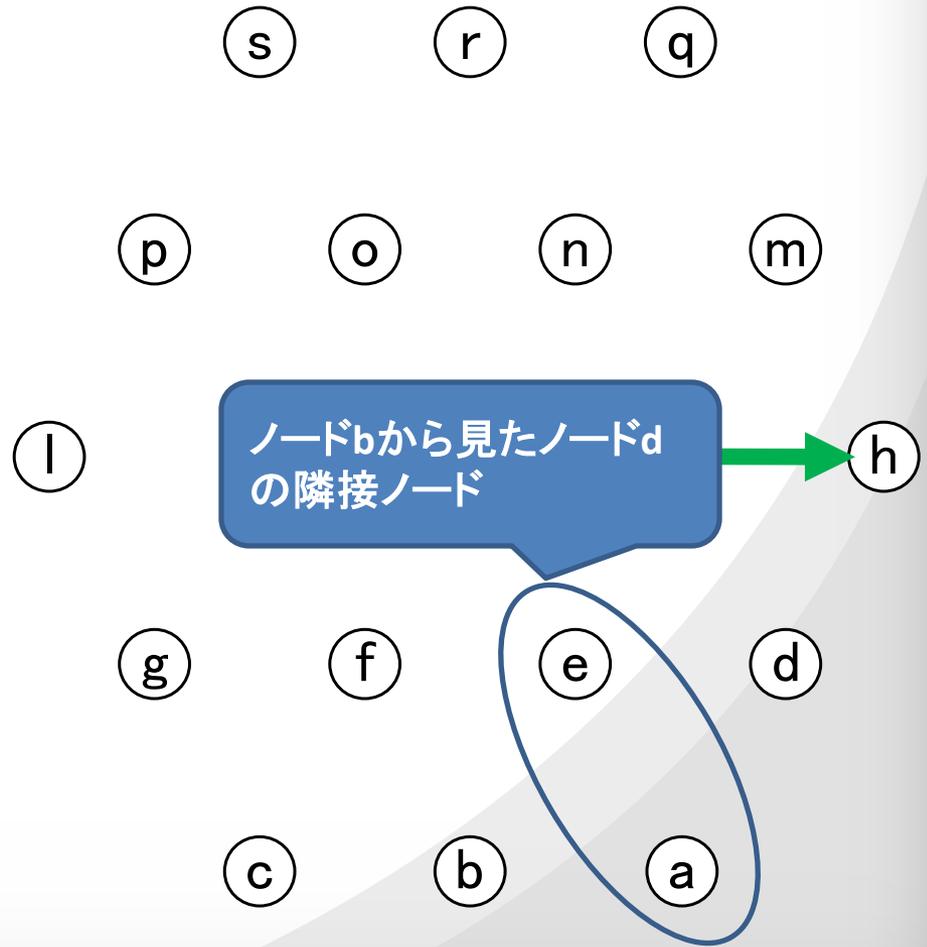
宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
省略	省略	省略
n	e	3
o	e	3
p	f	3
q	e	4
r	e	4
s	e	4



OLSRの経路生成

ノードbのRT

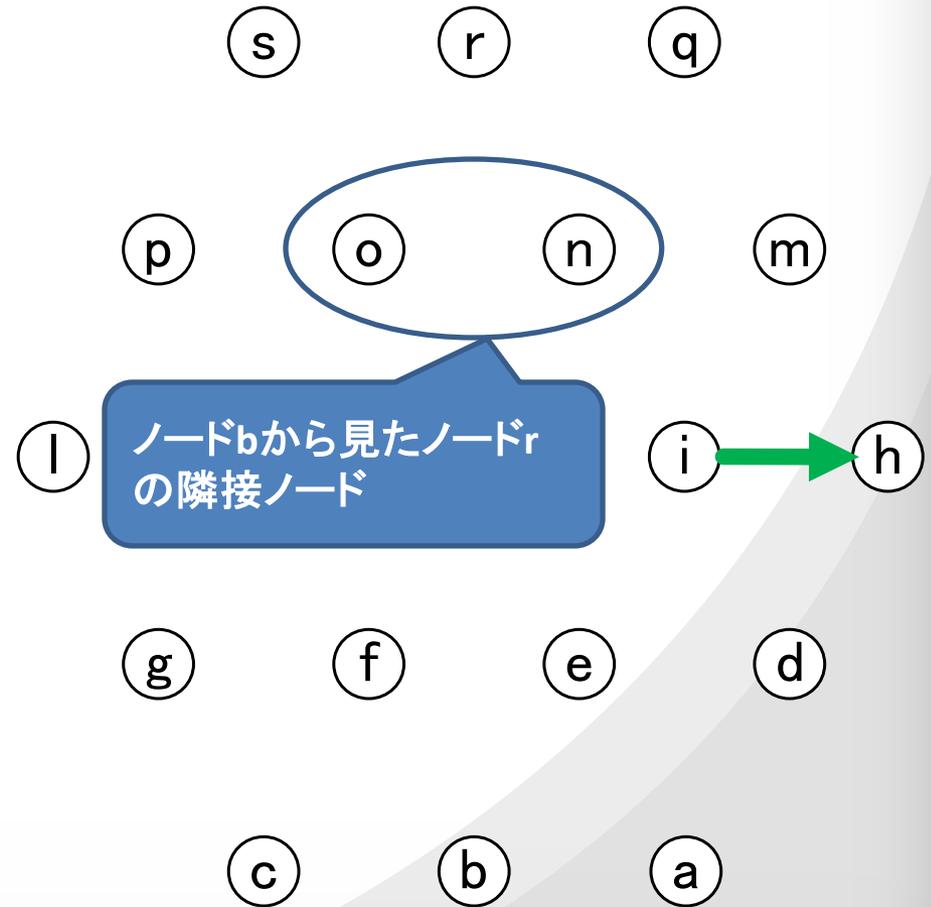
宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
省略		
n	e	3
o	e	3
p	f	3
q	e	4
r	e	4
s	e	4



OLSRの経路生成

ノードbのRT

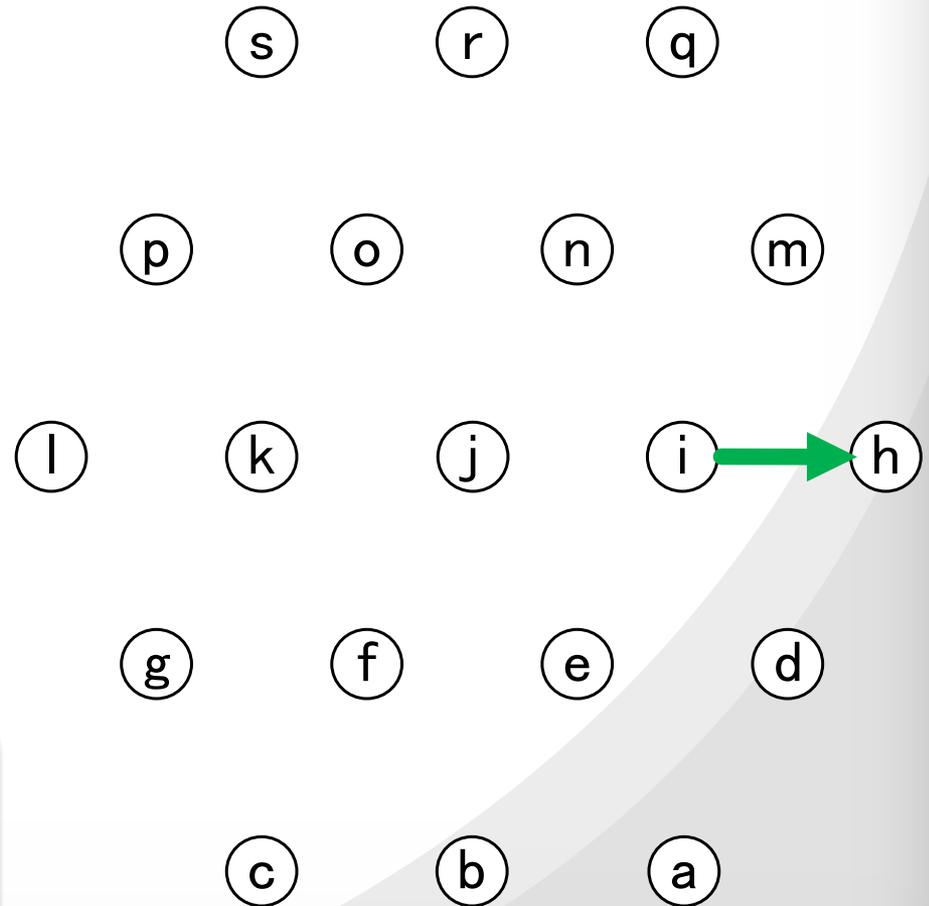
宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
省略		
n	e	3
o	e	3
p	f	3
q	e	4
r	e	4
s	e	4



OLSRの経路生成

ノードbのRT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
省略	省略	省略
r	e	4
s	e	4

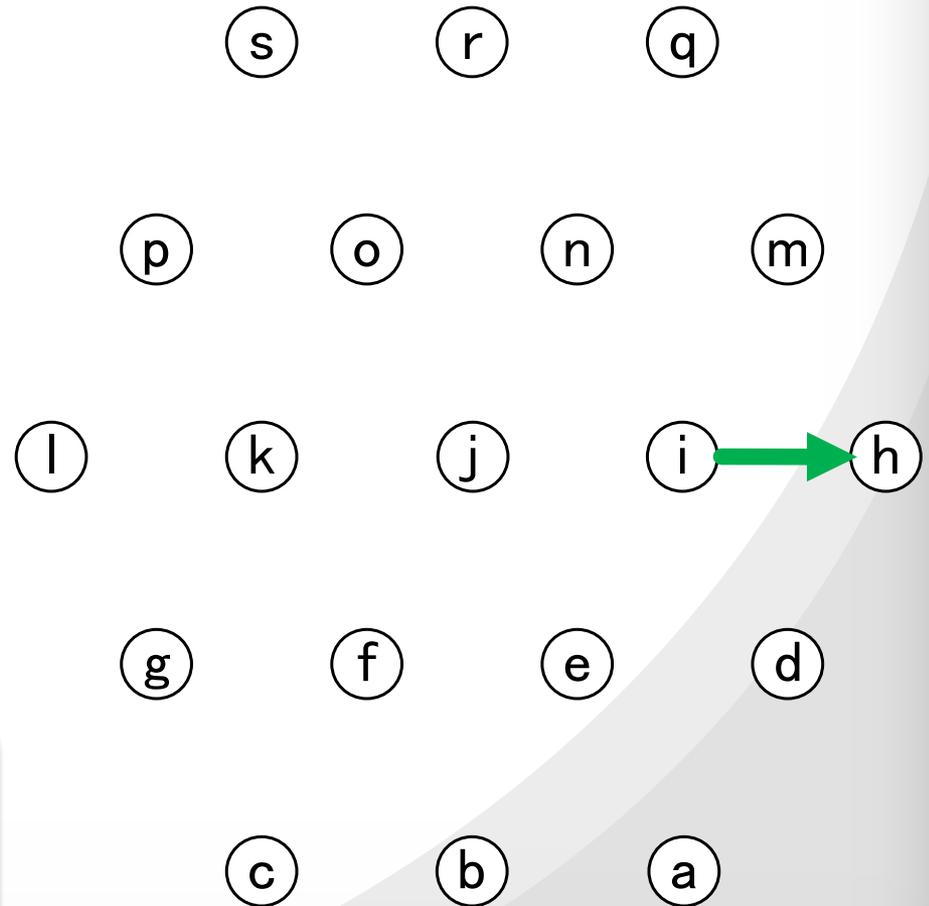


同様の方法で他のノードでもRT
が生成完了

OLSRの課題

ノードbのRT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
省略	省略	省略
r	e	4
s	e	4



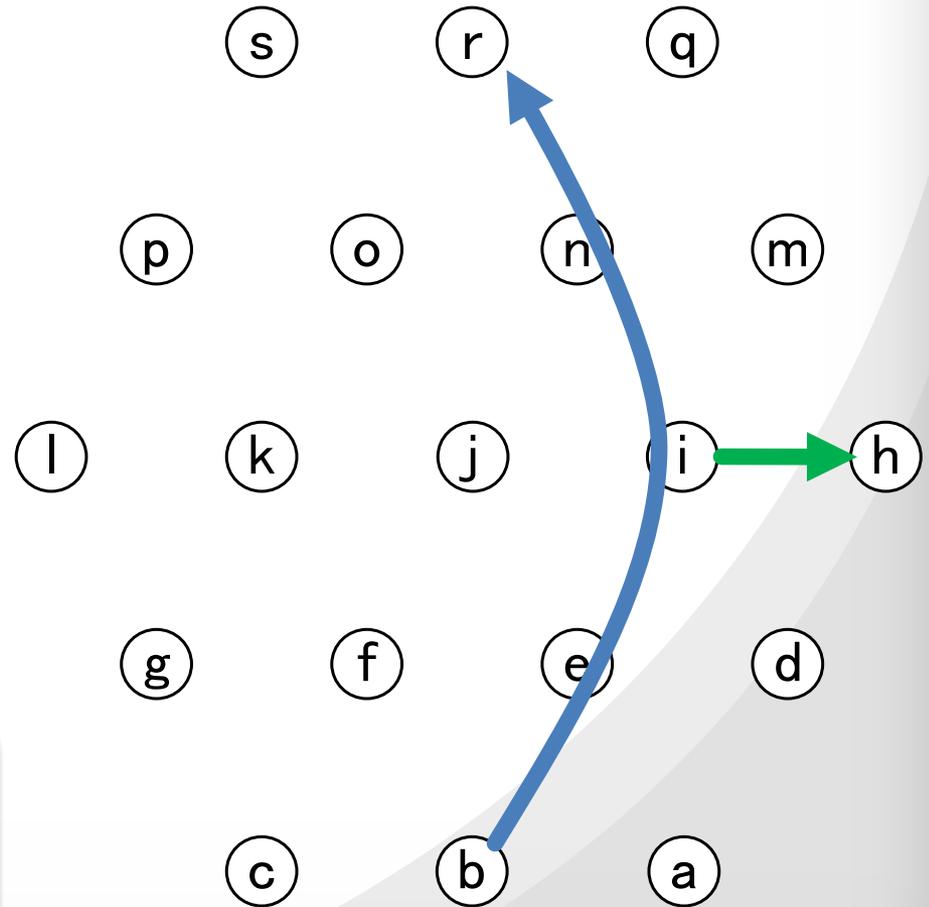
新たにノードbからノードrへ通信が発生

OLSRの課題

ノードbのRT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
~~~~~	省略	~~~~~
r	e	4
s	e	4

新たにノードbからノードrへ通信が発生

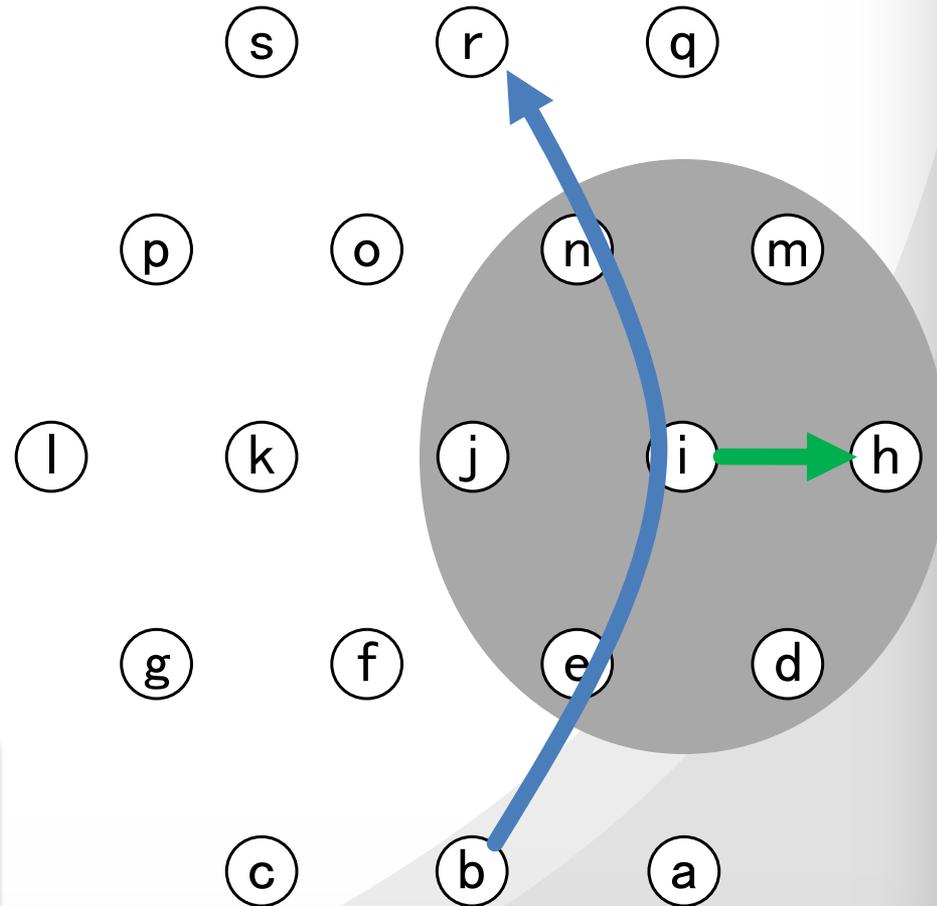


# OLSRの課題

ノードbのRT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
省略	省略	省略
r	e	4
s	e	4

ノードiから送信されるパケットは  
全隣接ノードが検出する

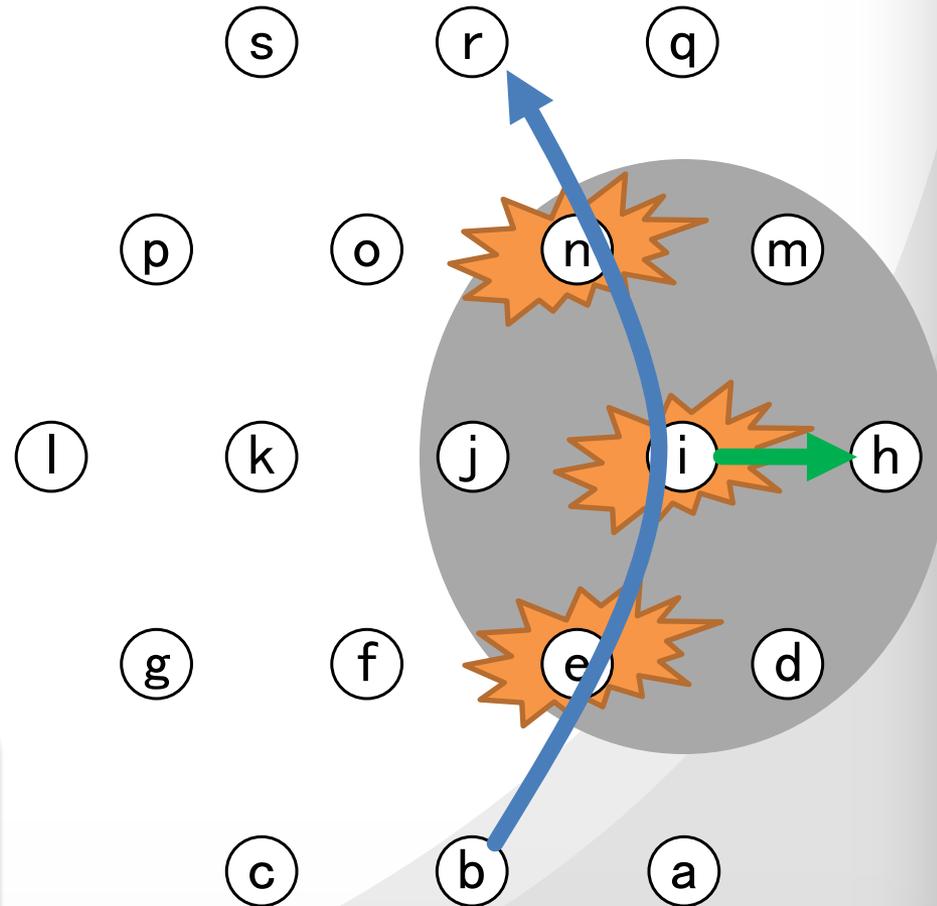


# OLSRの課題

ノードbのRT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
e	e	1
~~~~~	省略	~~~~~
r	e	4
s	e	4

ノードe, ノードi, ノードnでパケットロスが発生する可能性が高い



OLSRの課題 まとめ

- OLSR の経路選択は実装に依存しており, ネットワークのトラヒック状態を考慮したものとなっていない



- 新たなトラヒ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

課題の解決方法として, OLSRをベースとしたプロトコル PD-OLSR (Protocol Dependent-OLSR)を提案

提案方式

PD-OLSRの概要

- OLSRの基本部分はそのまま
- 各ノードの通信状態を考慮
- トランスポート層のTCPとUDPで専用のRTを生成することによって、効率の良い通信を実現

PD-OLSRの経路選択指標

- TCP通信 とUDP通信の特性の違いに着目

TCP・UDPの特性の違い

□UDP

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

輻輳制御によって順調にACKが返ってこればウィンドウサイズを拡大し帯域を使い切ろうとする

□混在するネットワークのトラフィック

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

PD-OLSRの経路選択指標

- UDP通信の経路選択指標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通信の経路選択指標TCP Session
 - 自身が検出するTCPセッション数の合計

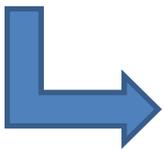
プロトコル毎のRT生成方法

- ① 各ノードは計測したUDP Traffic , TCP SessionをHELLO, TCメッセージに乗せて隣接ノードへ広告
- ② 各ノードは受信したHELLO, TCメッセージを基に新たに定義したRMT (Routing Metric Table) を生成
 - RMT は宛先ノード, 宛先への次ホップノード, ホップ数, 経路選択指標 (次ホップノードの UDP Traffic, TCP Session) から構成され, 複数の最短経路候補を有する
- ③ RMTを基にTCP通信用RTとUDP通信用RTを生成
 - RMT からUDP通信用RTに選ばれる経路
 - UDP Trafficが最小の経路
 - RMT からTCP通信用RTに選ばれる経路
 - TCP の特性を活かし帯域幅の均等性がとれるように, TCP Sessionが最小の経路
 - TCP Sessionが同じであった場合は、UDP Trafficの少ない経路

ノードbのRMT

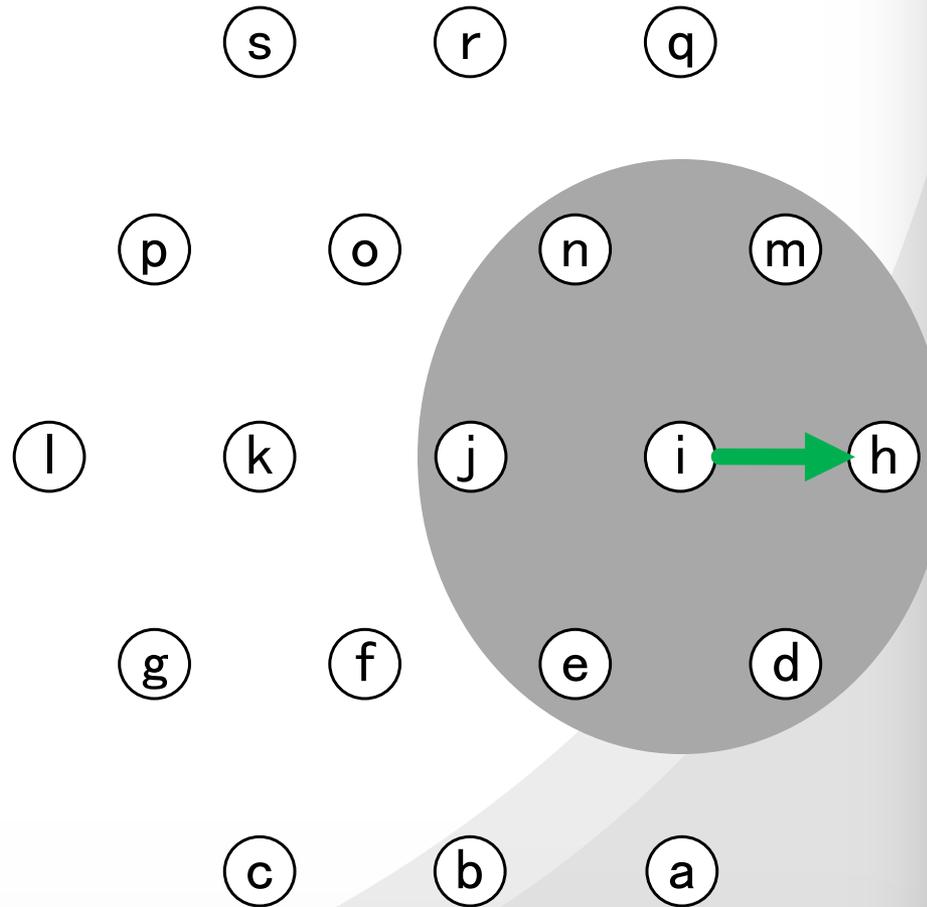
宛先	次ホップ	ホップ数	UDP Traffic
a	a	1	0
c	c	1	0
d	a	2	0
省略			
r	e	4	8
r	f	4	0
s	e	4	8
s	f	4	0

ノードbのUDP通信用RT



宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
省略		
r	f	4
s	f	4

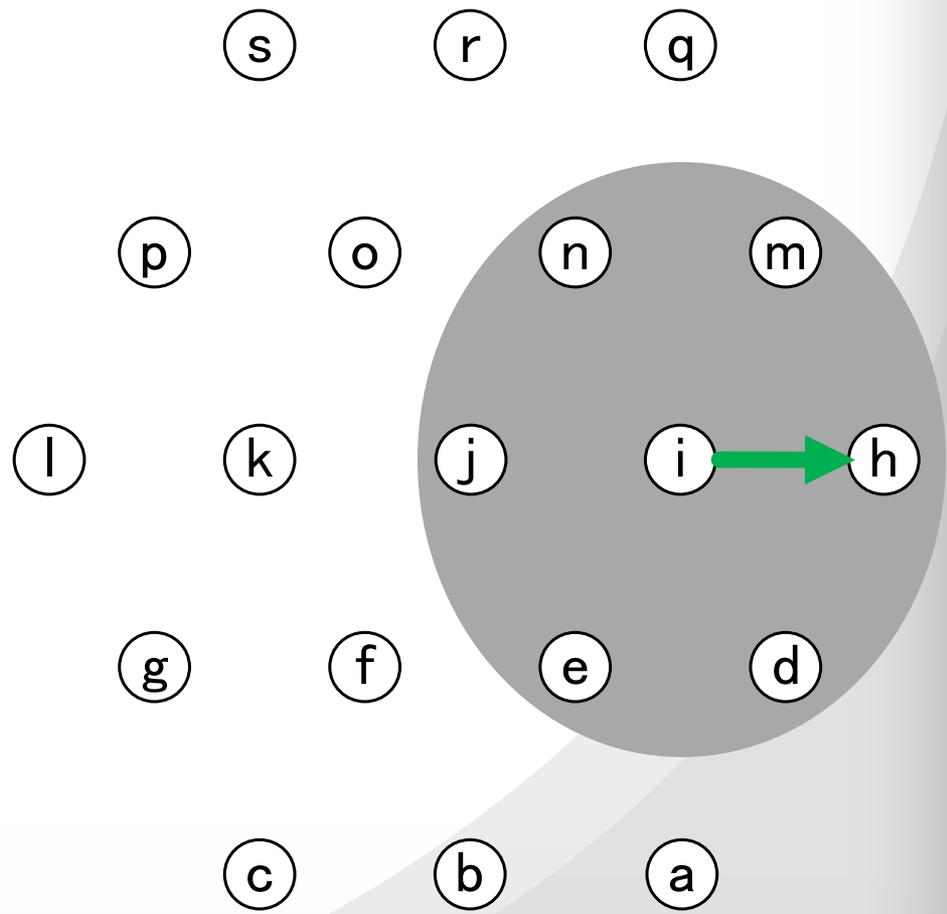
UDP通信用RTの生成



ノードbのRMT

宛先	次ホップ	ホップ数	UDP Traffic
a	a	1	0
c	c	1	0
d	a	2	0
省略			
r	e	4	8
r	f	4	0
s	e	4	8
s	f	4	0

UDP通信用RTの生成



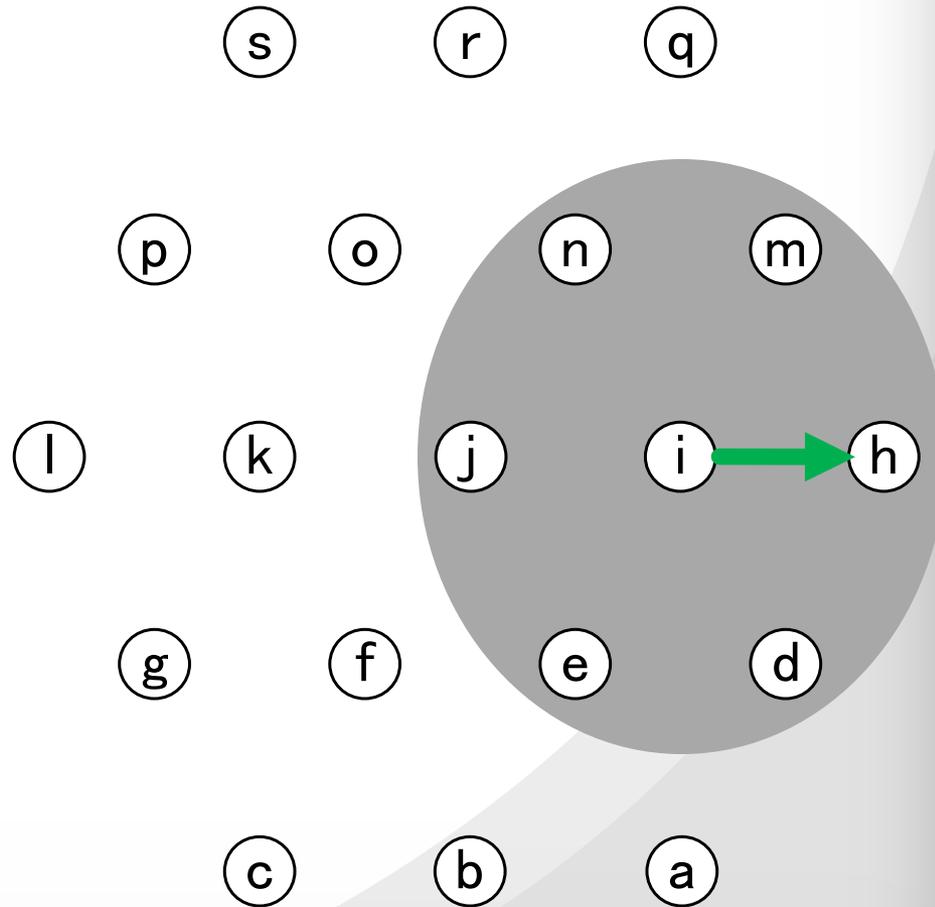
ノードbのUDP通信用RT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
省略		
r	f	4
s	f	4

ノードbのRMT

宛先	次ホップ	ホップ数	UDP Traffic
a	a	1	0
c	c	1	0
d	a	2	0
省略			
r	e	4	8
r	f	4	0
s	e	4	8
s	f	4	0

UDP通信用RTの生成



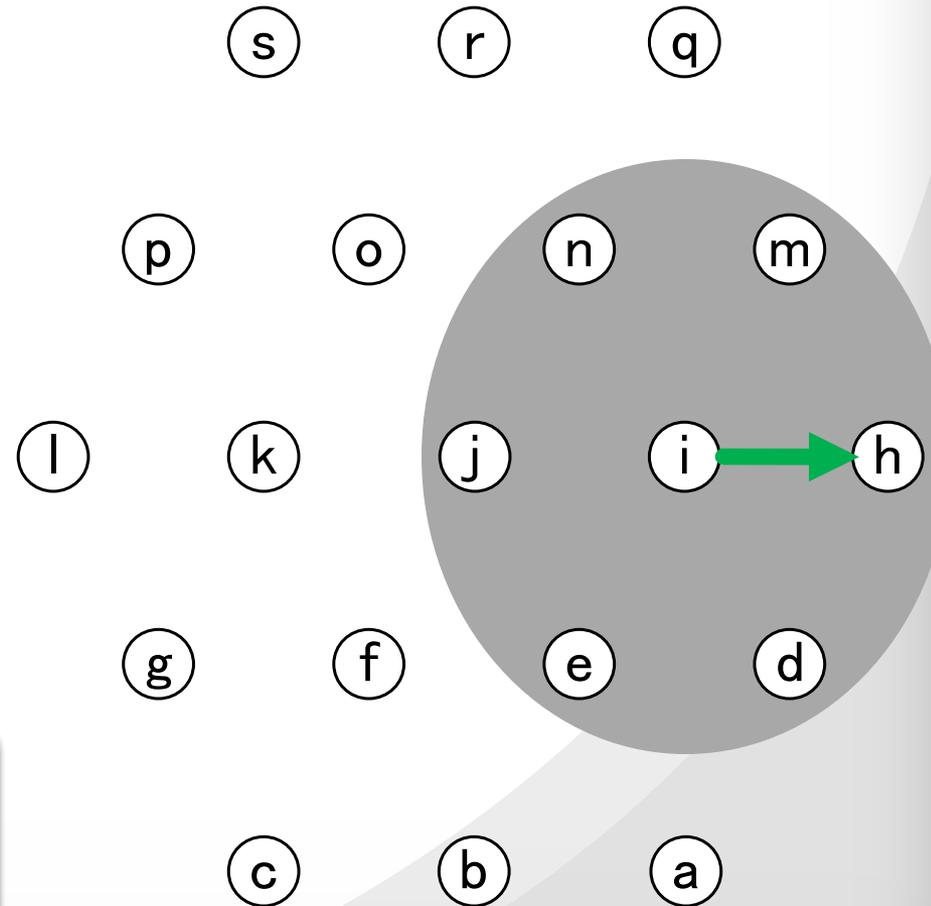
ノードbのUDP通信用RT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
省略		
r	f	4
s	f	4

UDP通信用RTの生成

ノードbのUDP通信用RT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
省略		
r	f	4
s	f	4

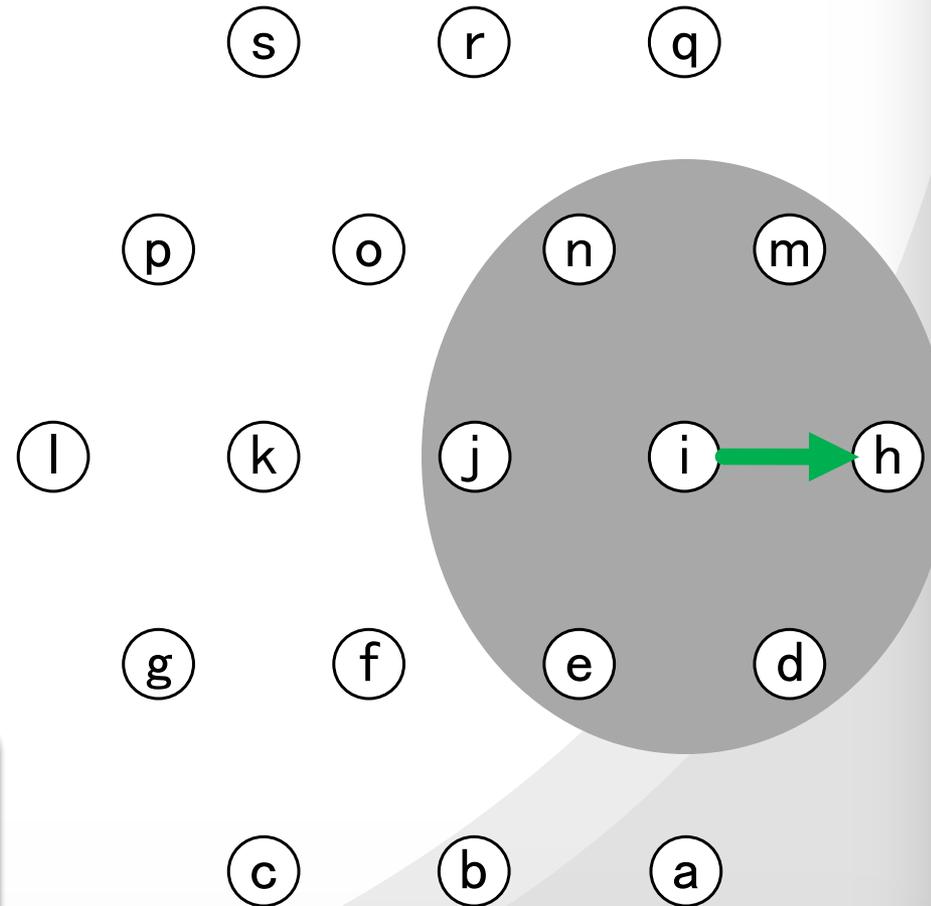


同様にして、全ノードでUDP通信用RTを生成が完了

UDP通信用RTの生成

ノードbのUDP通信用RT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
省略		
r	f	4
s	f	4



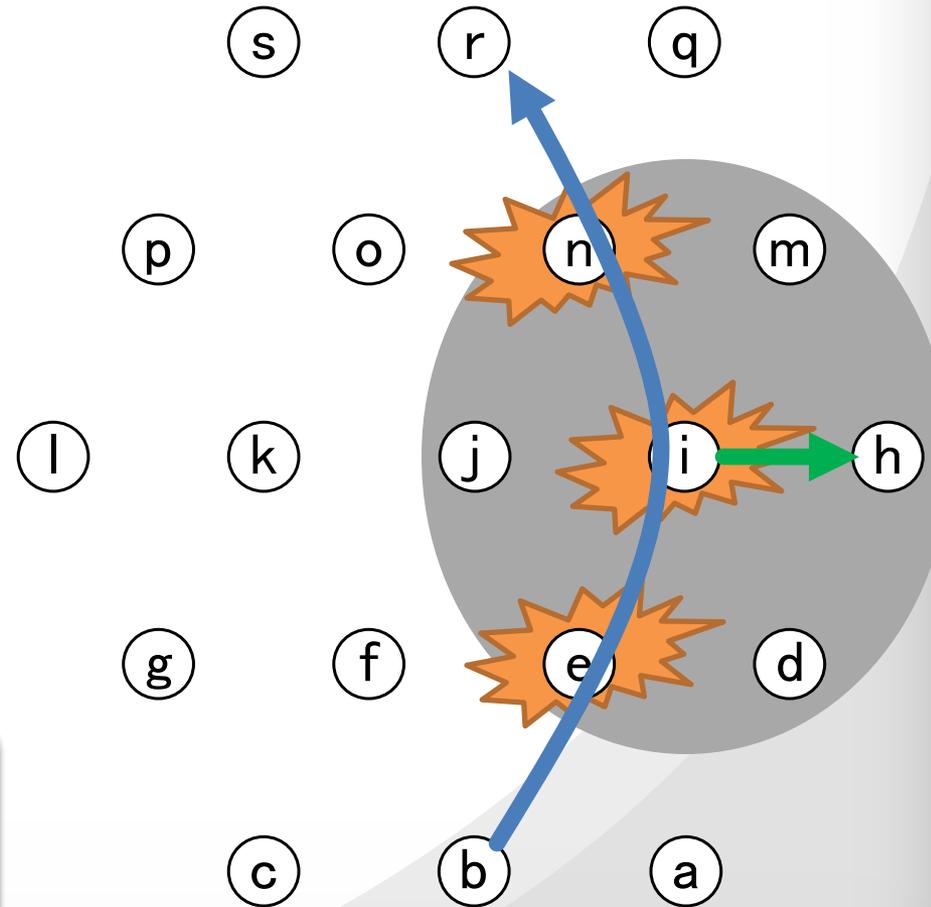
新たにノードbからノードrへ通信が発生

UDP通信用RTの生成

ノードbのUDP通信用RT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
~~~~~ 省略 ~~~~~		
r	f	4
s	f	4

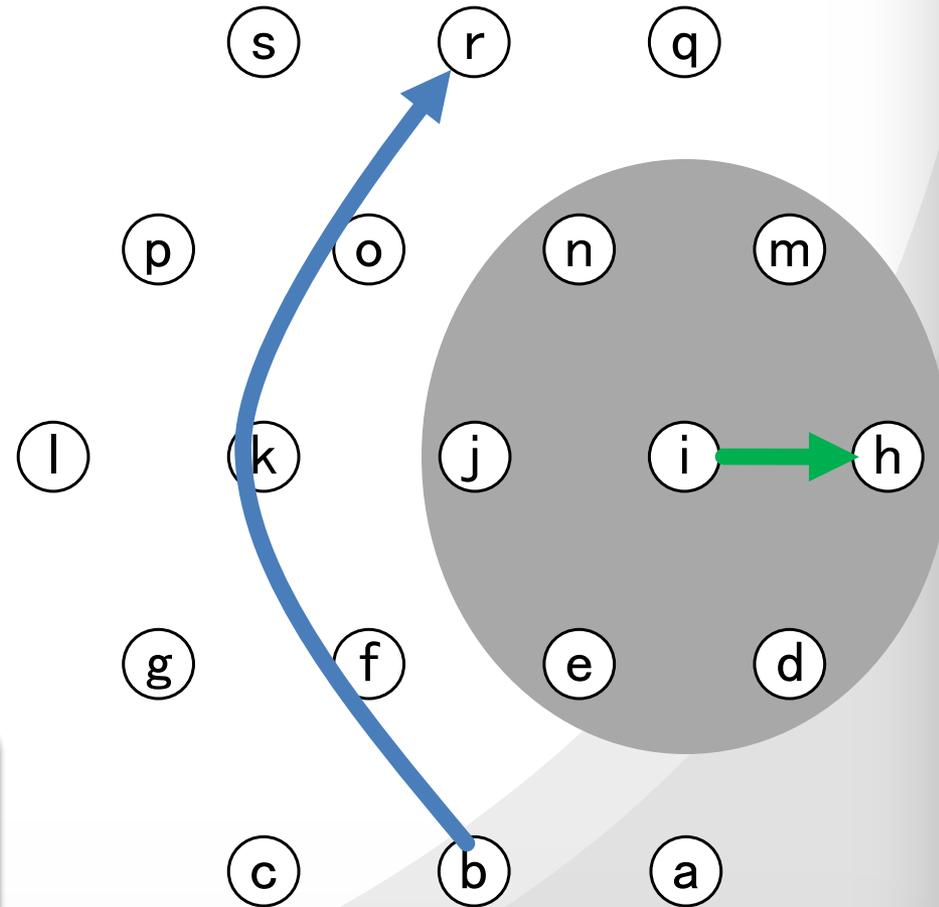
新たにノードbからノードrへ通信が発生



# UDP通信用RTの生成

## ノードbのUDP通信用RT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
~~~~~ 省略 ~~~~~		
r	f	4
s	f	4

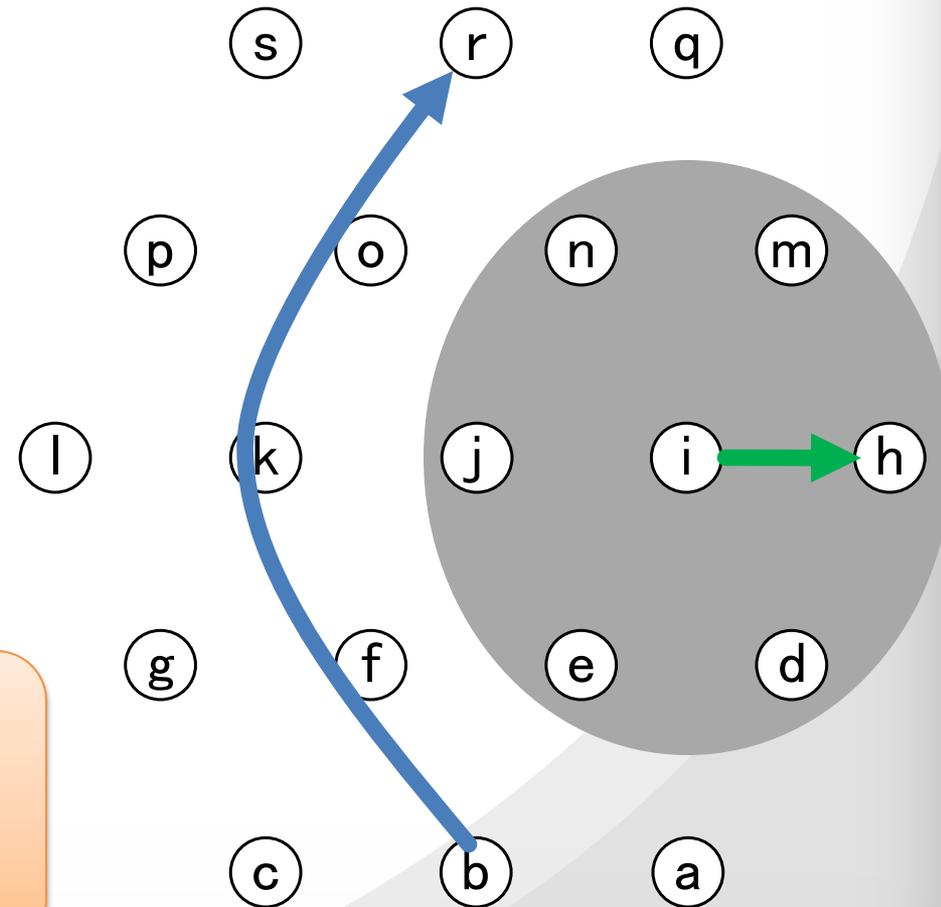


ノードiからノードhへの通信のトラフィックの影響を受けない経路で通信が行える

UDP通信用RTの生成

ノードbのUDP通信用RT

宛先	次ホップ	ホップ数
a	a	1
c	c	1
d	a	2
~~~~~ 省略 ~~~~~		
r	f	4
s	f	4



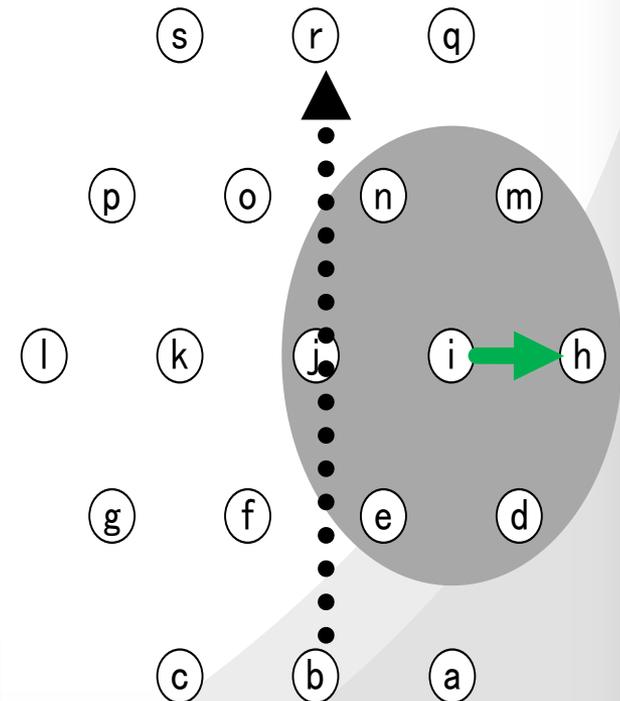
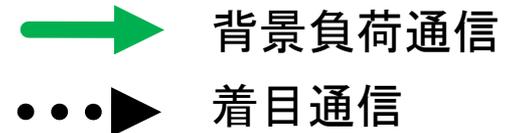
TCP通信においても、RMTからRTに経路を選択する過程でTCP Sessionも考慮すれば、帯域幅の均等性がとれる経路で通信が行える

# 評価

# 動作検証

## ● 環境

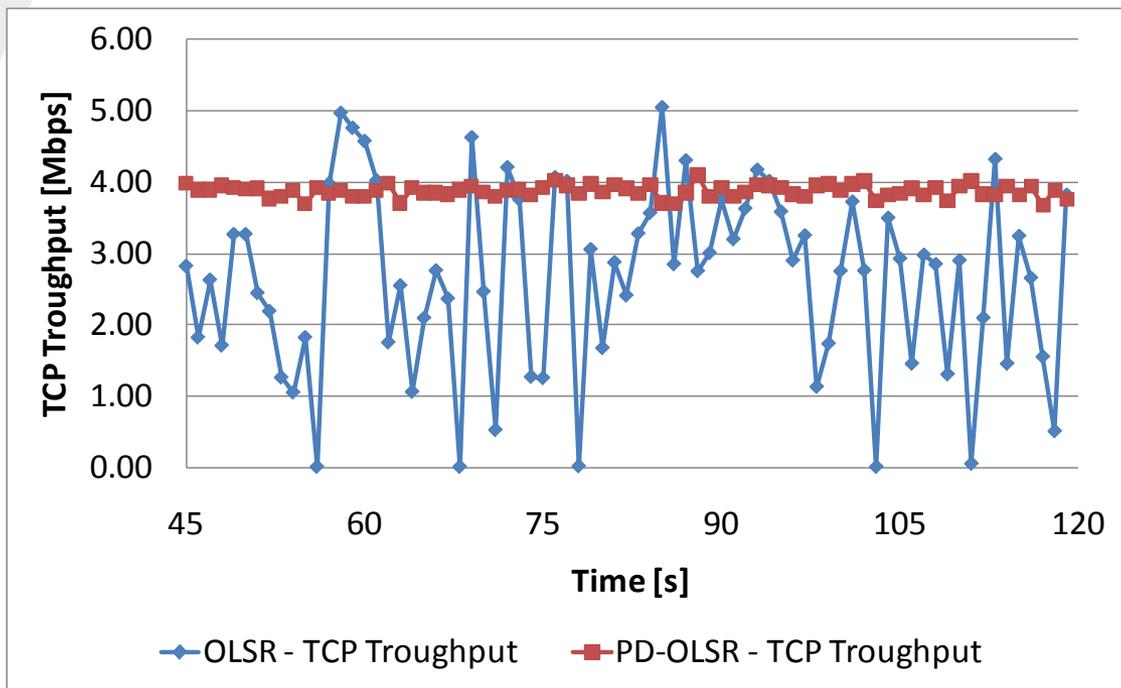
アドホック ネットワーク	ノード数	19 [台]
	電波到達範囲	100 [m]
	ノード間距離	95 [m]
	ルーティングプロトコル	OLSR, PD-OLSR
背景負荷通信 ノードi→ノードh	無線規格	802.11g
	通信タイプ	CBR
	トランスポートプロトコル	UDP
	データ転送量	1 [Mbps]
着目通信 ノードb→ノードr	ノードi→ノードh	200 [Byte]
	パケットサイズ	1 [Mbps]
	通信タイプ	FTP
	トランスポートプロトコル	TCP
ノードb→ノードr	パケットサイズ	1000 [Byte]
ノードb→ノードr	最大衝突ウィンドウサイズ	20 [pkt]



- シミュレーション開始30秒後に背景負荷通信を開始させ、さらに15秒後に着目通信を開始
- 合計120秒間のシミュレーションを各プロトコルで行い、着目通信のTCPスループットを比較

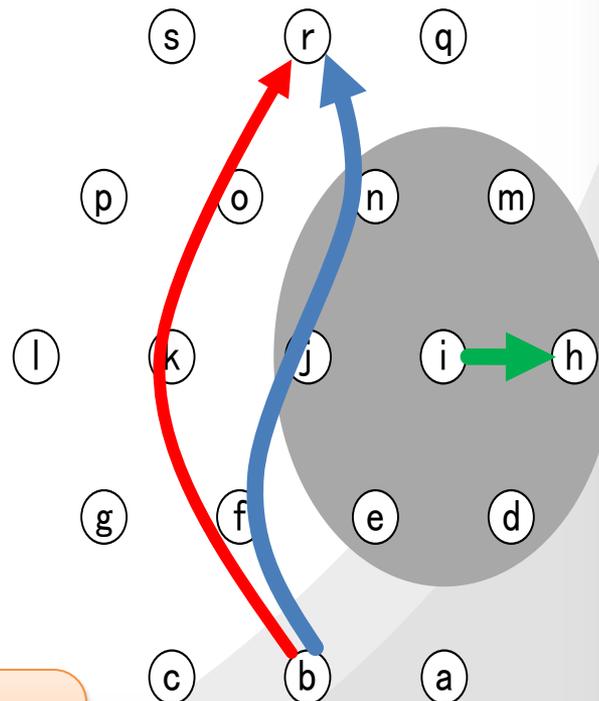
# 動作検証

## ● 結果



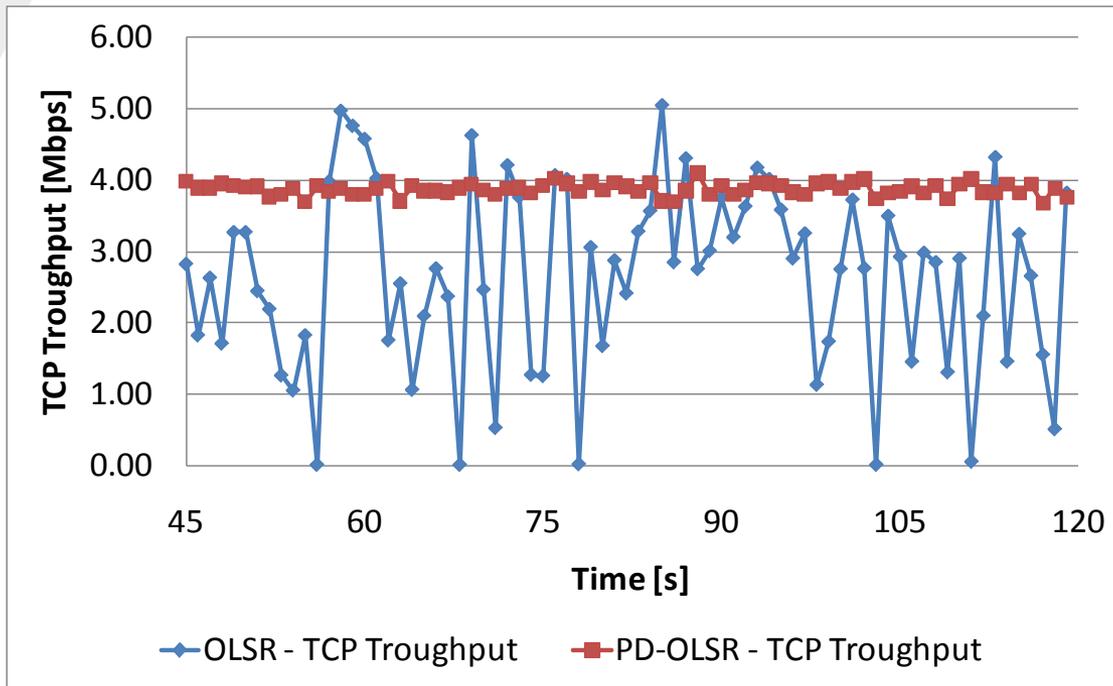
TCP スループットの平均は、OLSR では約 2.6Mbps, PD-OLSR では約3.8Mbps であり、約 1.5 倍の違いがあった

→ OLSRの場合の経路  
→ PD-OLSRの場合の経路

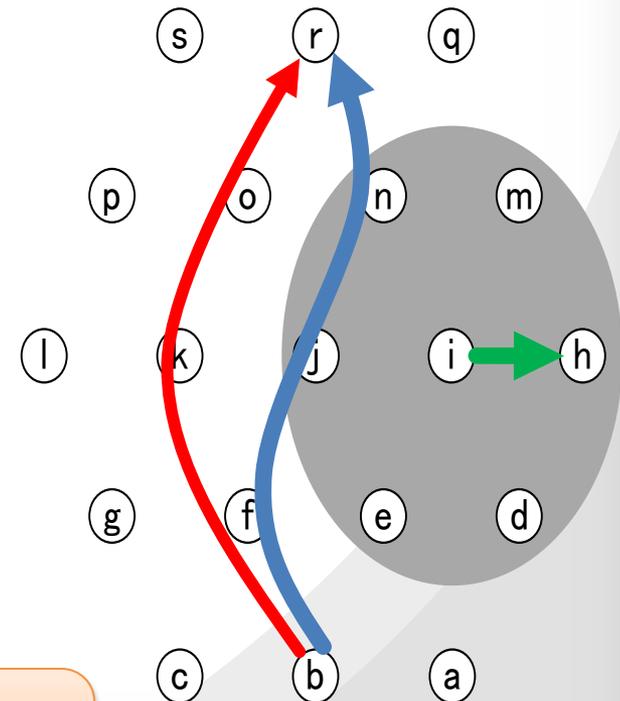


# 動作検証

## ● 結果



➡ OLSRの場合の経路  
➡ PD-OLSRの場合の経路

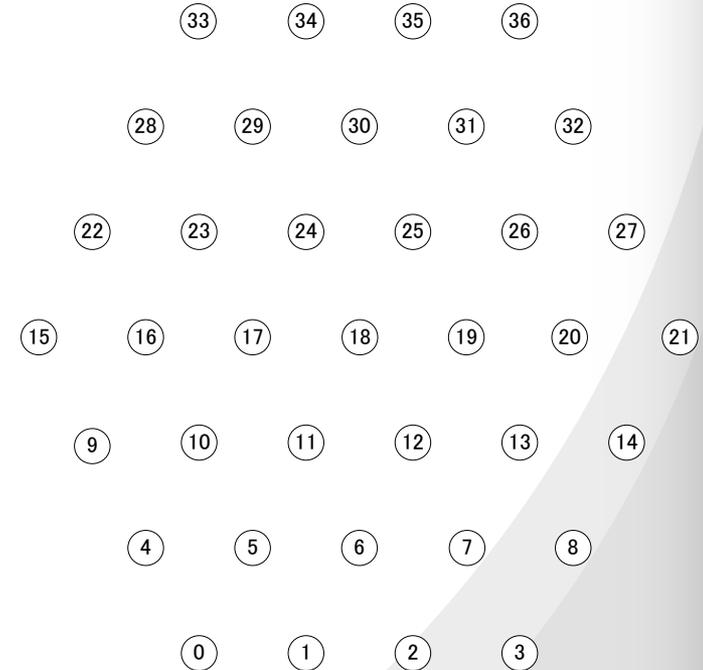


PD-OLSRが正しくシミュレータに実装されていることを確認できた

# 大規模シミュレーション

## ● 環境

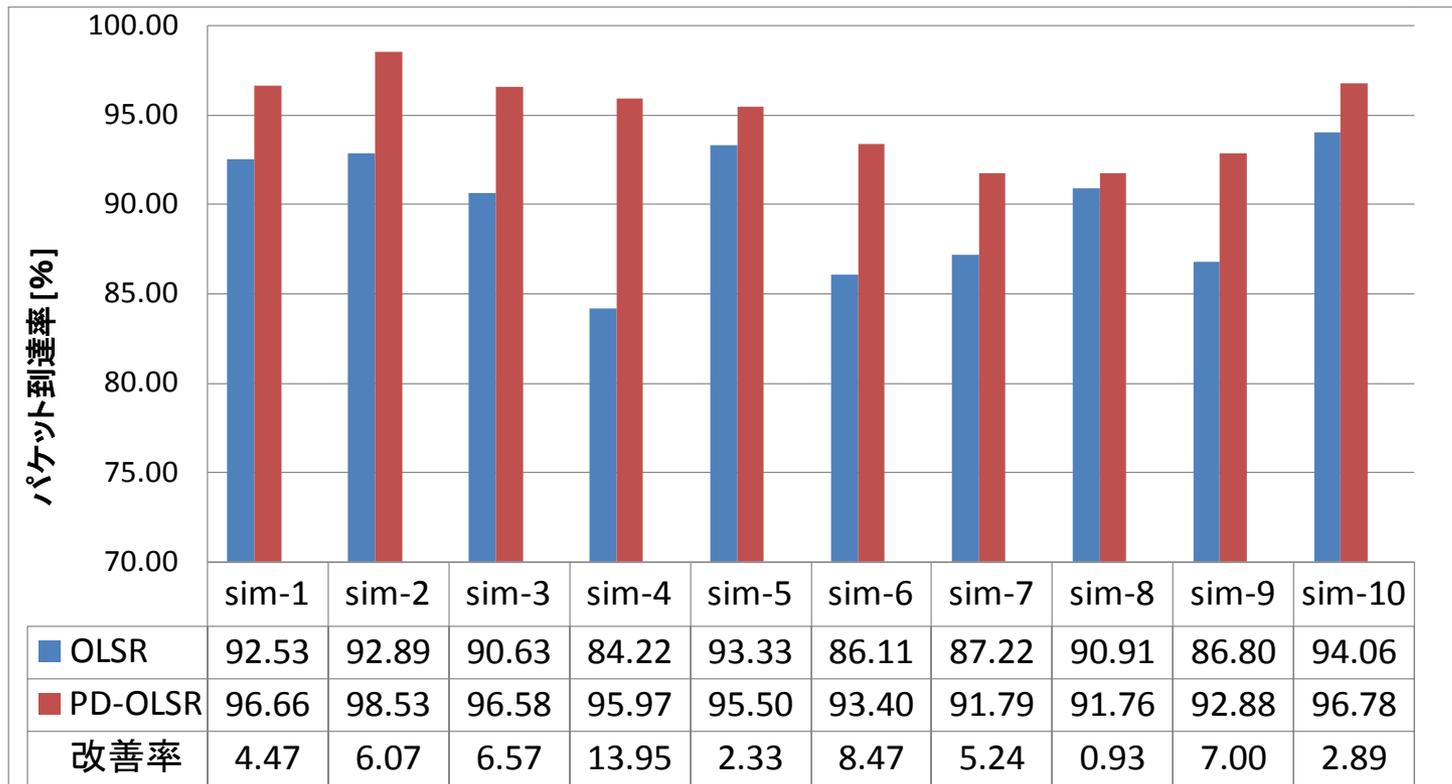
アドホック ネットワーク	ノード数	37 [台]
	電波到達範囲	100 [m]
	ノード間距離	95 [m]
	ルーティングプロトコル	OLSR, PD-OLSR
VoIPを想定した UDP通信	無線規格	802.11g
	台数	2台1ペア
	選び方	ランダム
	通信タイプ	CBR
	トランスポートプロトコル	UDP
	パケットサイズ	200 [Byte]
データ転送量	64 [Kbps]	



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

# 大規模シミュレーション

## ● 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

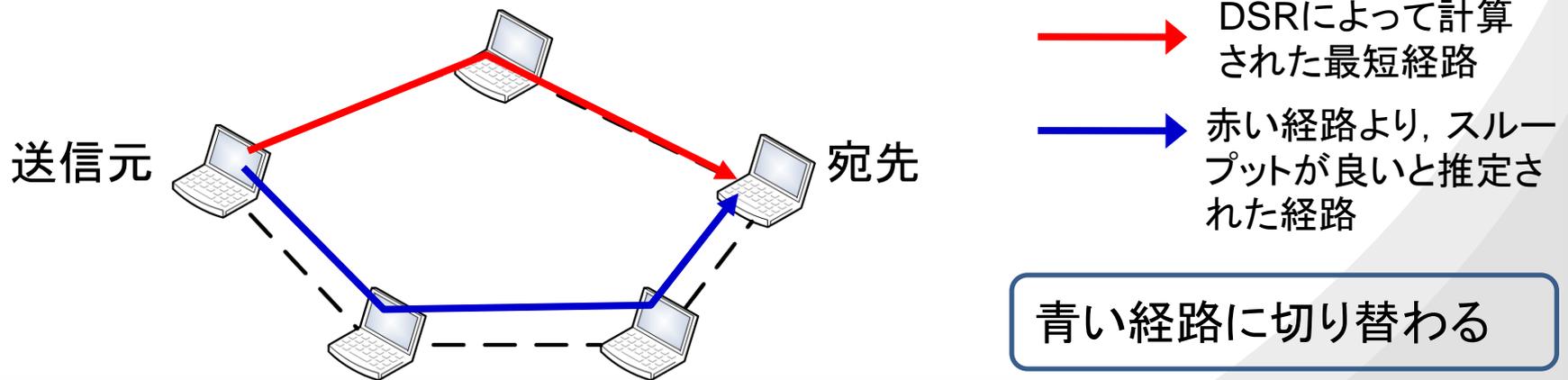
- 本発表
  - OLSRを拡張することによって, TCP用とUDP用のRTを別々に生成し, 経路上の通信状態を考慮して経路を生成できるプロトコルPD-OLSRを提案
  - UDP通信用のRT生成機能をシミュレータに実装し, VoIP通信を想定したシミュレーションを行った
  - 結果として, UDP通信においてはトラフィックの高い経路を避けた通信を行うことによって, パケット到達率が6%程度向上することが分かった
- 今後の予定
  - TCP用のRT生成機能をシミュレータに実装し, 動作検証を行う
  - 他のプロトコルに提案方式の機能を実装した場合や, 新たな経路選択指標と合わせて経路生成が行える方法を検討する

**ご清聴ありがとうございました**

# 補足

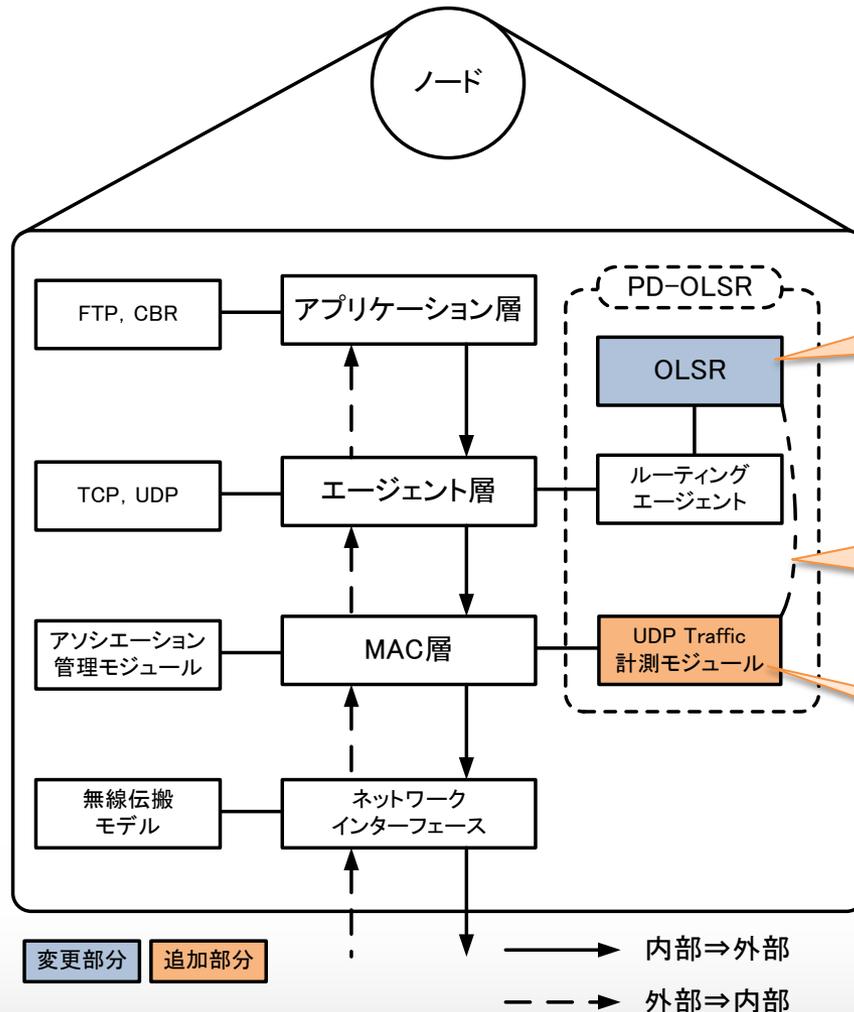
# 関連研究

- ETR (Estimated-TCP-Throughput Maximization based Routing)
  - リアクティブ型のDSR (Dynamic Source Routing)を拡張
  - DSRによって計算された最短経路から, スループットが良いと推定される経路に切り替える
  - スループットの算出に必要な遅延と往復パケット損失率は, データ送信開始時から一定間隔で送信されるRTPLM (Round-Trip Packet Loss ratio Measurement) 要求とその応答によって収集



一定間隔でRTPLM要求の送信を行うため,  
ネットワークへのオーバーヘッドが高くなる

# シミュレータへの実装



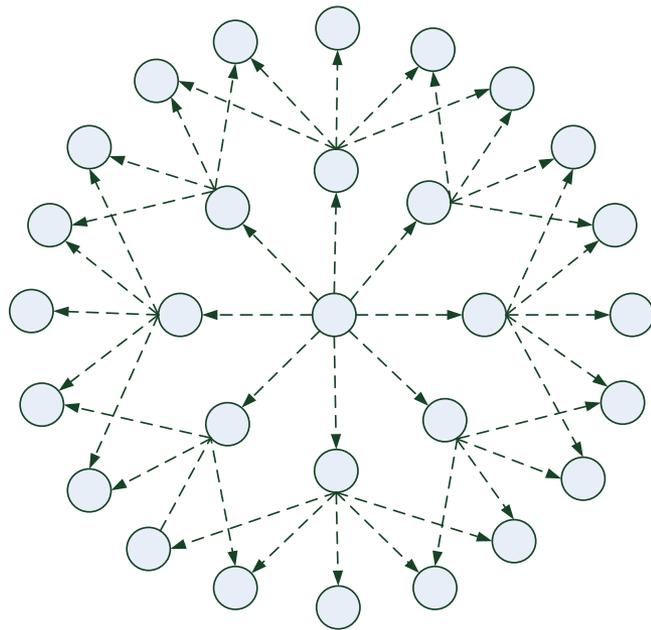
UDP 通信用のRT生成機能をns2  
に実装した

③ OLSRを PD-OLSR の経路生成  
動作が行えるように拡張した

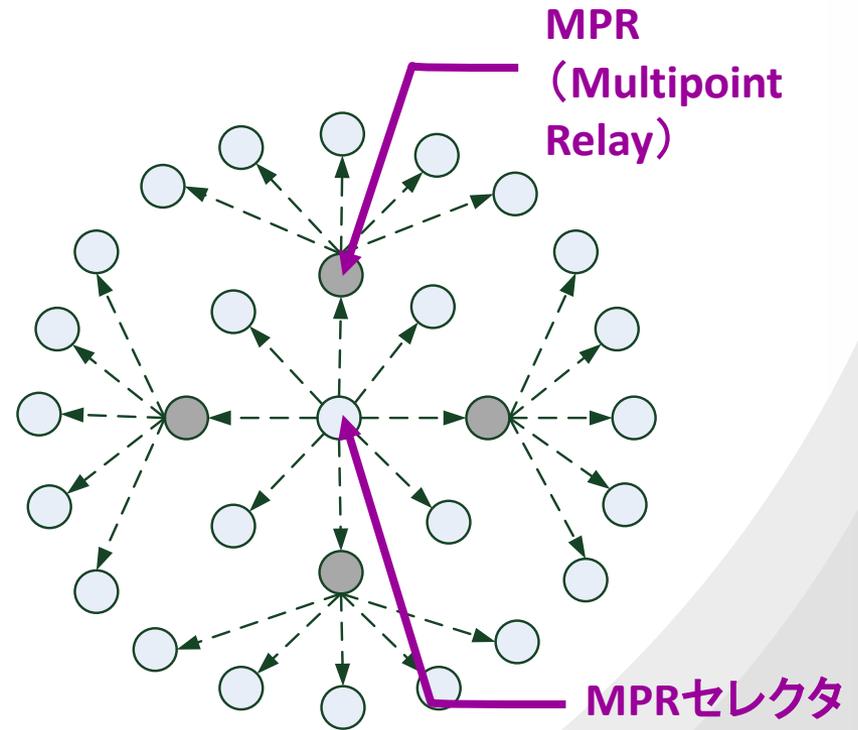
② UDP Traffic 計測モジュール  
で計測した UDP Trafficをルー  
ティングエージェントで呼び出  
せるようにした

① UDP Traffic を計測するモ  
ジュールを追加した

# OLSRのフラッディング

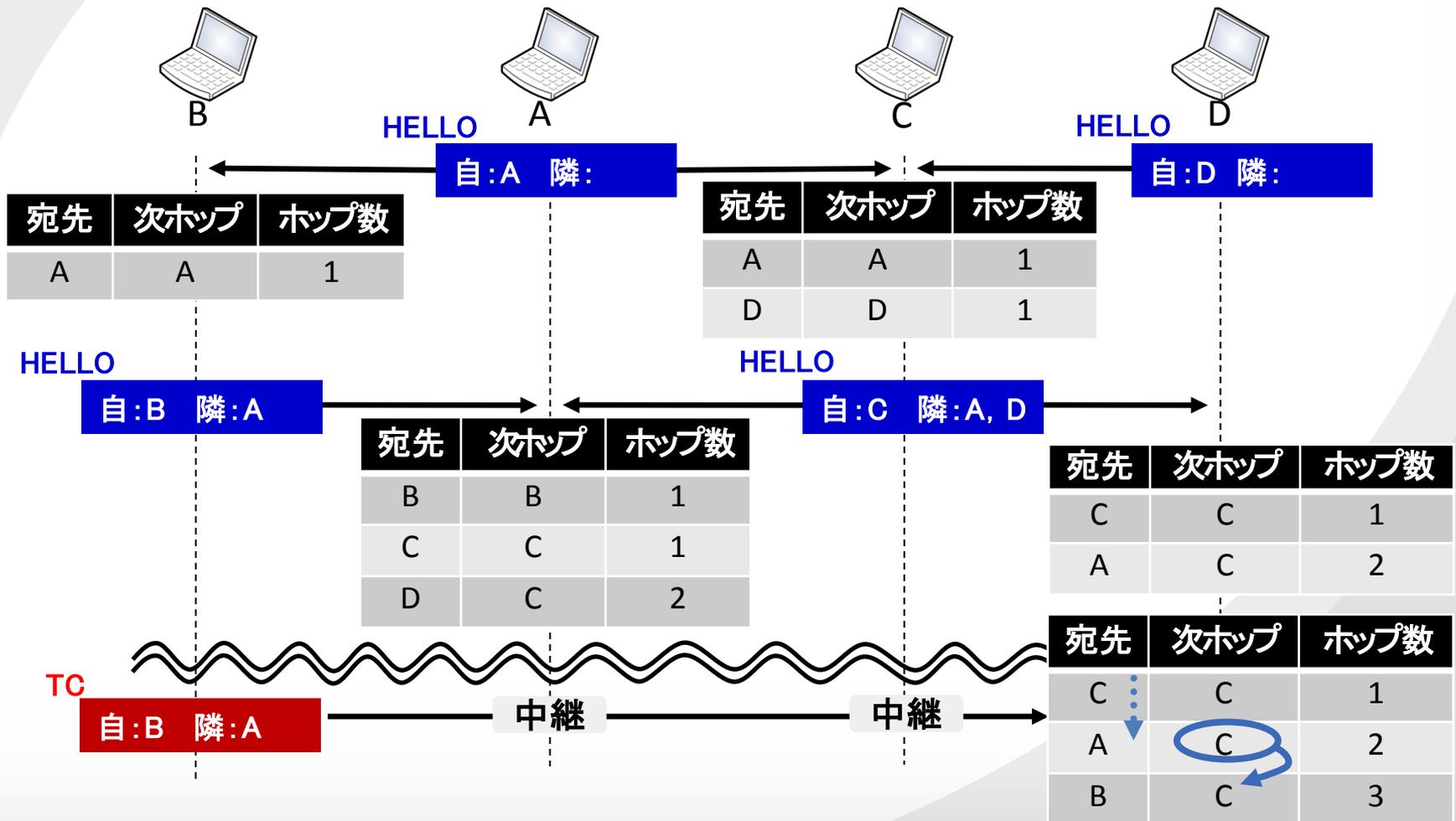


通常のフラッディング

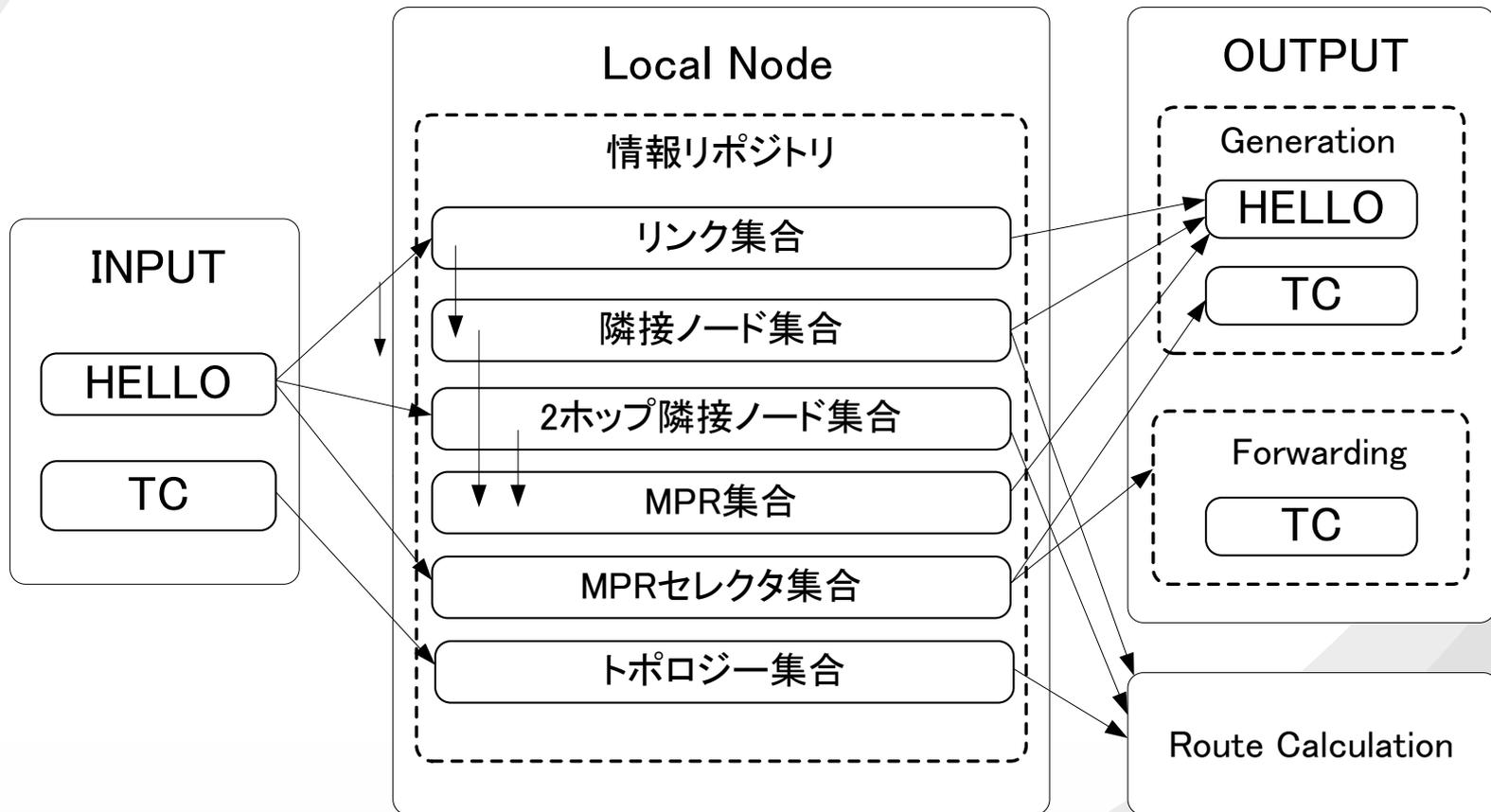


OLSRのフラッディング

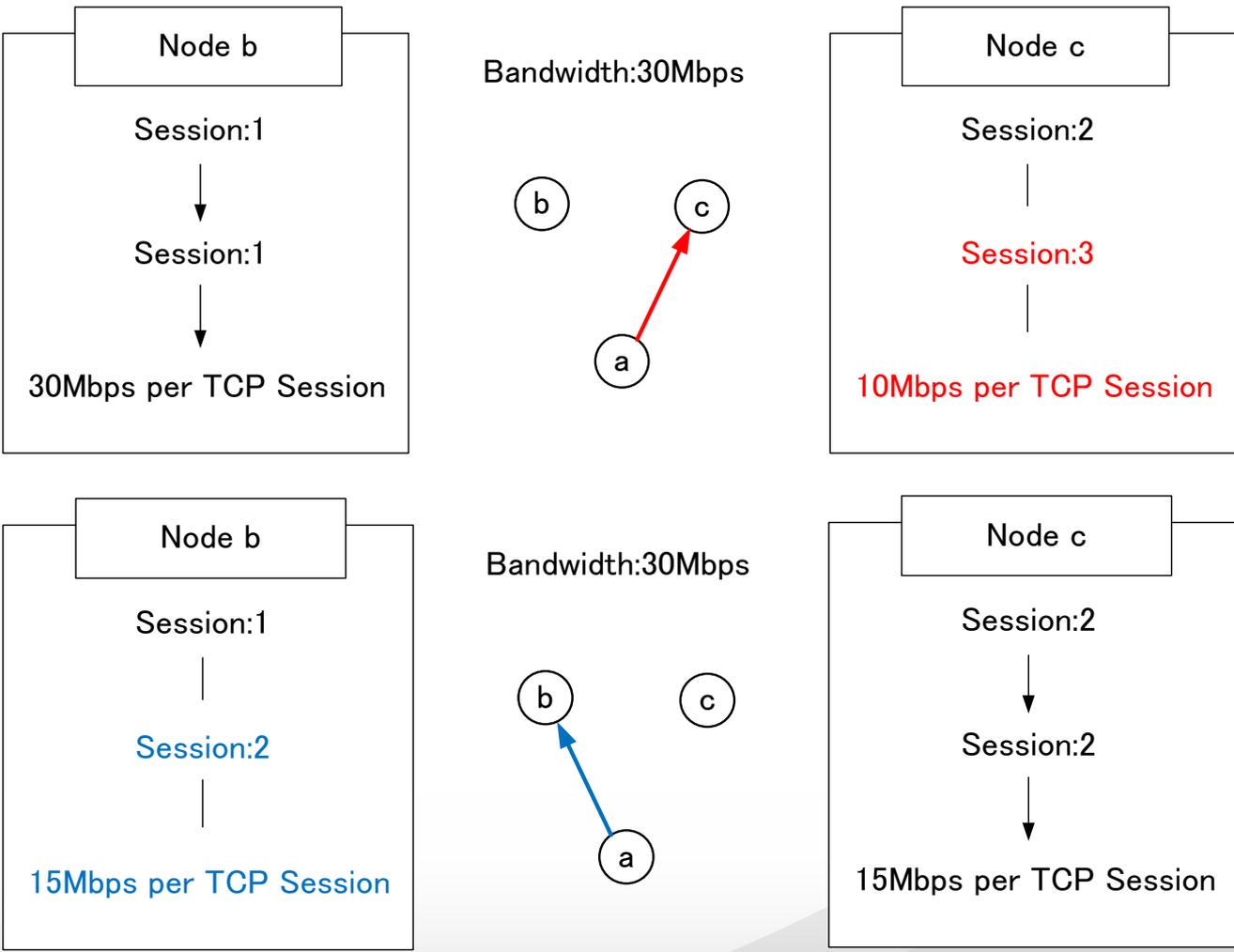
# OLSRのRT生成方法



# ノード内部でのOLSRの動作



# 帯域幅の均等性



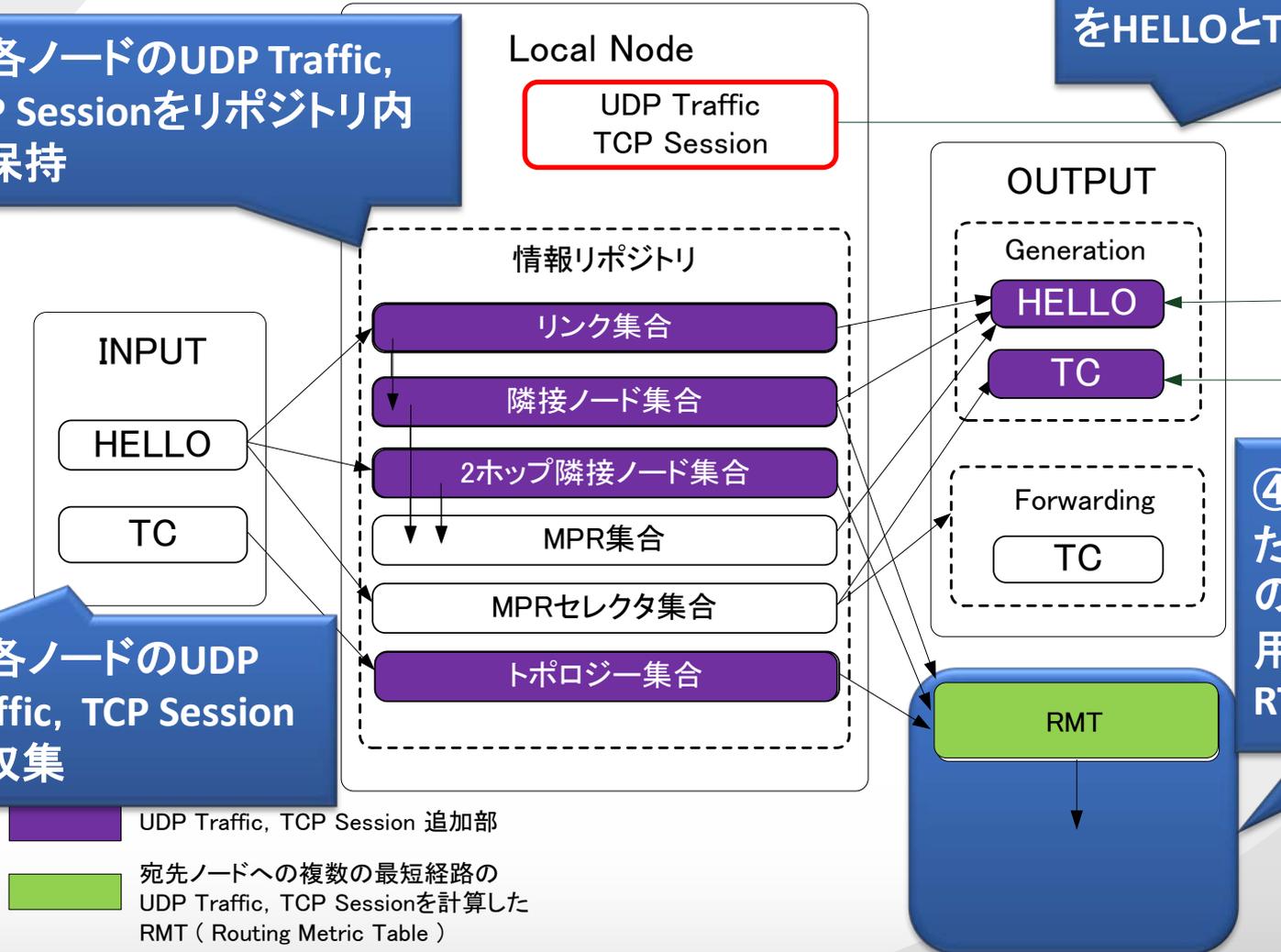
# OLSRの拡張方法

① UDP Traffic, TCP Session  
をHELLOとTCに付加

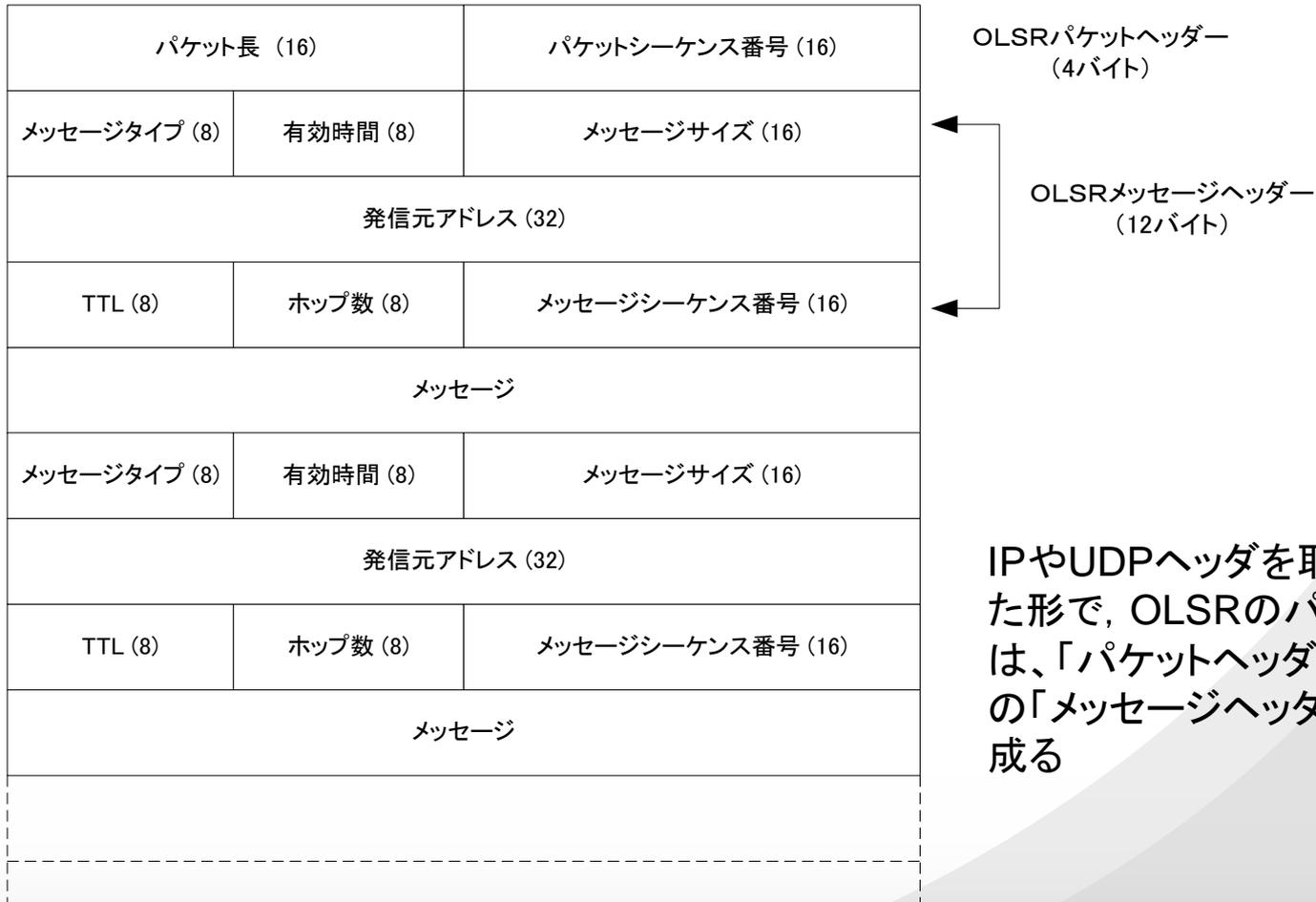
③各ノードのUDP Traffic,  
TCP Sessionをリポジトリ内  
で保持

②各ノードのUDP  
Traffic, TCP Session  
を収集

④RMTで生成され  
た複数の最短経路  
の中からUDP・TCP  
用の経路を選択し、  
RTを生成



# OLSRのパケットフォーマット



IPやUDPヘッダを取り除いた形で、OLSRのパケットは、「パケットヘッダ」と複数の「メッセージヘッダ」から成る

# HELLOメッセージフォーマット

予約 (16)		HELLO発生間隔 (8)	Willingness (8)
リンクコード (8)	予約 (8)	リンクメッセージサイズ (16)	
隣接ノードのインタフェースアドレス (32)			
隣接ノードのインタフェースアドレス (32)			
⋮			
隣接ノードのインタフェースアドレス (32)			
隣接ノードのインタフェースアドレス (32)			
リンクコード (8)		予約 (8)	リンクメッセージサイズ (16)
隣接ノードのインタフェースアドレス (32)			
隣接ノードのインタフェースアドレス (32)			
⋮			

OLSR_HELLOヘッダー

OLSR_HELLOメッセージヘッダー

1メッセージ

# TCメッセージフォーマット

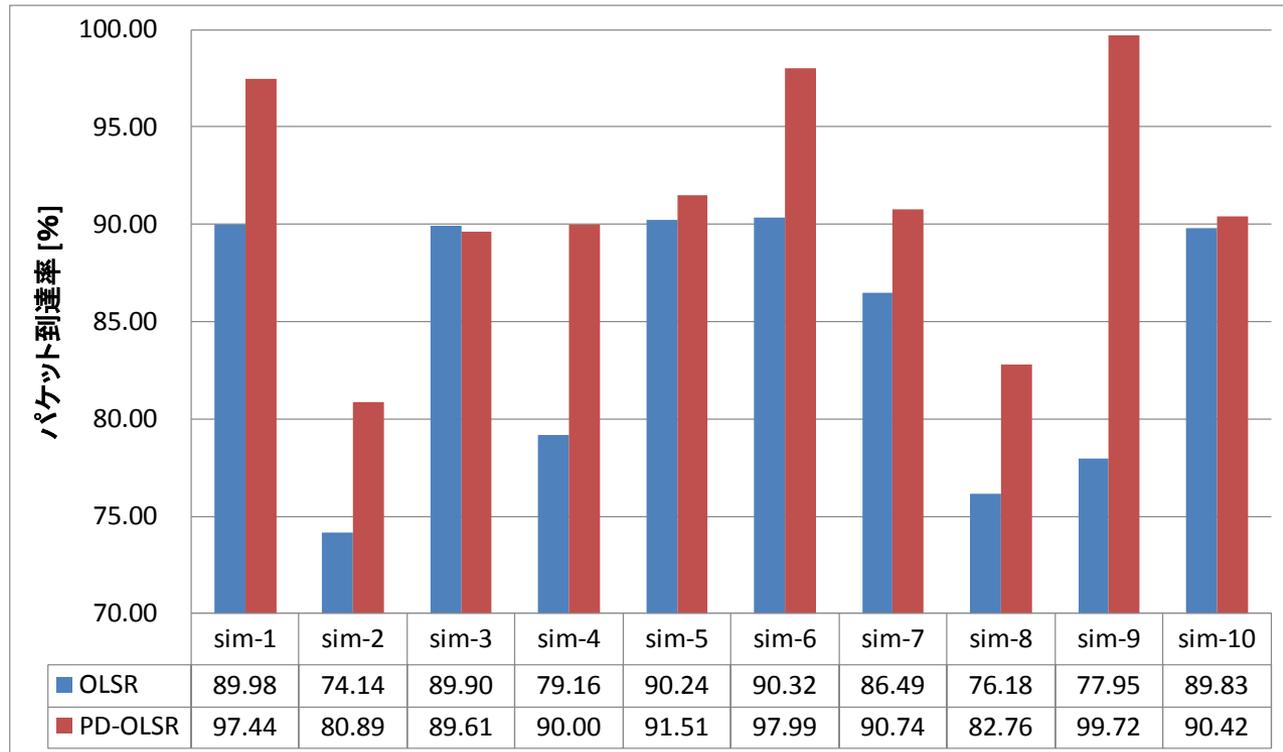
近隣広告シーケンス番号 (16)	予約 (16)	OLSR_TCヘッダー
近隣広告アドレス (32)		
近隣広告アドレス (32)		
.		
-----		



# 大規模シミュレーション

## ● 結果

2秒間隔毎に  
セッションを追  
加した場合



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8%改善された

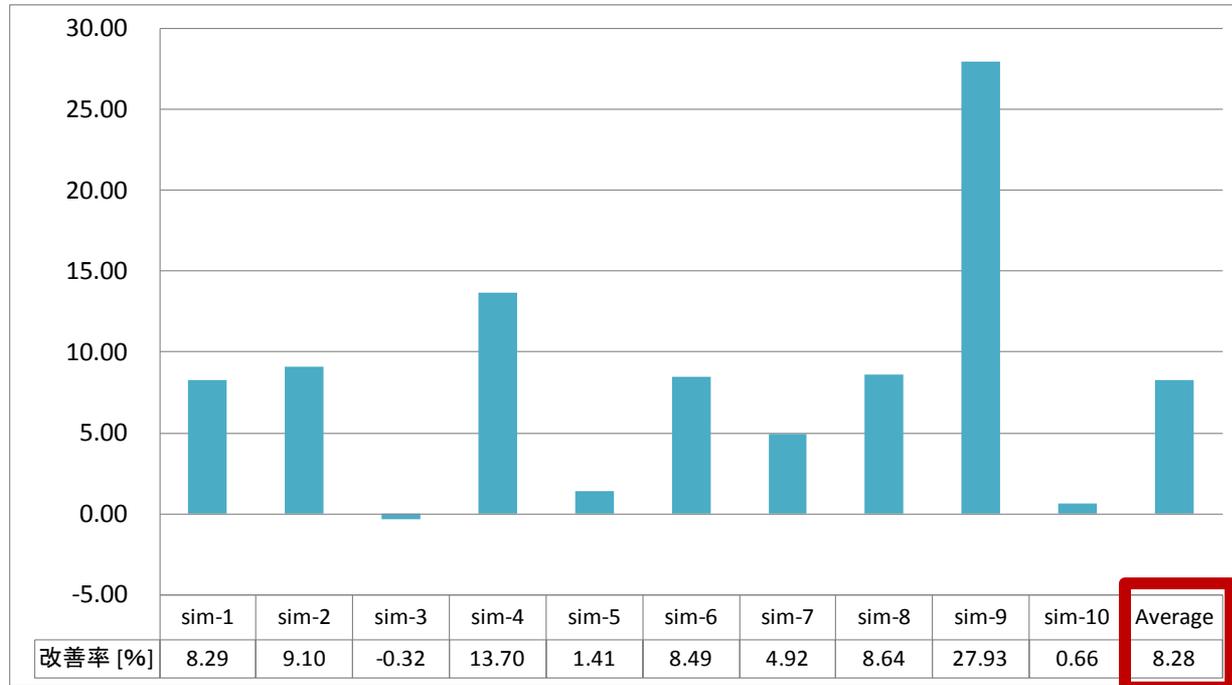
- まだ使えるかもしれないスライド

# 本発表の構成

- 研究背景
  - MANET
  - アドホックルーティングプロトコルの課題
- OLSR
  - 経路生成方法
  - OLSRの課題
- 提案方式
  - PD-OLSRの概要
  - PD-OLSRの経路選択指標
  - PD-OLSRの経路生成方法
- 評価
  - シミュレータへの実装
  - 動作検証
  - 大規模シミュレーション
- まとめ

# 大規模シミュレーション

## 結果②



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約8%改善された

# アドホックルーティングプロトコルの課題

- 多くのアドホックルーティングプロトコルは、中継ホップ数が最小となる最短経路を選択する
- 最短経路が複数存在する場合はどの経路を選択するかは実装に任されている

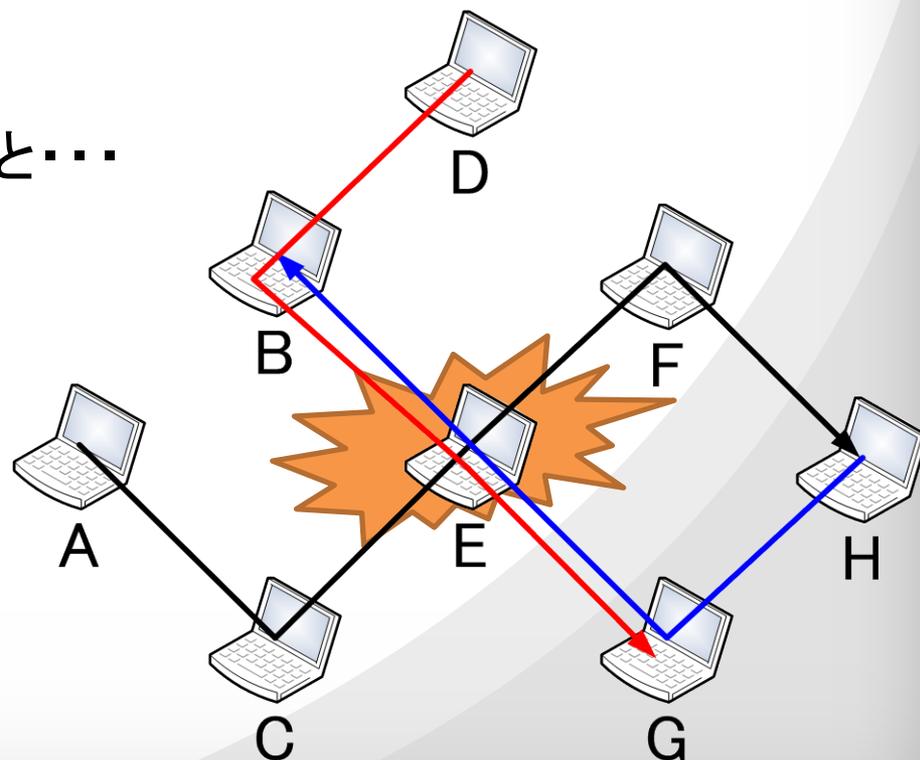
# アドホックルーティングプロトコルの課題

- 多くのアドホックルーティングプロトコルは、中継ホップ数が最小となる最短経路を選択する
- 最短経路が複数存在する場合はどの経路を選択するかは実装に任されている
  - トラフィックが集中するノードを通る経路で通信が行われると…

パケットロスが多発



スループットが低下



# PD-OLSRの概要

- OLSRの基本部分はそのまま
- 各ノードで経路選択指標を求める
  - UDP通信用の経路選択指標UDP Traffic
    - 自身が検出するネットワーク上のキャリアの合計
  - TCP通信用の経路選択指標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
- 各ノードは経路選択指標(UDP Traffic, TCP Session)をOLSRの制御メッセージに乗せて隣接ノードへ広告
- 各ノードは受信した制御メッセージを基に新たに定義した経路計算テーブル(RCT:Route Calculation Table)を生成
  - RCTは宛先ノード(Dest), 宛先への次ホップノード(Next), ホップ数(hop), 経路選択指標(NextのUDP Traffic, NextのTCP Session)から構成され, 複数の最短経路候補を有する
- TCP通信用RTとUDP通信用RTをRCTを基に生成
  - RCTからUDP通信用RTに選ぶ経路
    - UDPトラフィックが最小の経路
  - RCTからTCP通信用RTに選ぶ経路
    - 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セッション数が最小のノードを選ぶ

# PD-OLSRの概要

- OLSRの基本部分はそのまま
- TCP通信とUDP通信の特性を考慮した経路生成を行う
  - TCP通信用とUDP通信用のRTを生成
  - UDP通信用の経路に選ぶノード
    - UDPトラフィックが最小のノードを選ぶ
  - TCP通信用の経路に選ぶノード
    - 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セッション数が最小のノードを選ぶ
- 各ノードで求めた経路選択指標(UDP Traffic, TCP Session)をOLSRの制御メッセージに乗せて隣接ノードへ広告
- 各ノードは受信した制御メッセージを基に新たに定義した経路計算テーブル(RCT:Route Calculation Table)を生成
  - RCTは宛先ノード(Dest), 宛先への次ホップ, ノード(Next), ホップ数(hop), 通信状態情報(NextのUDP Traffic, NextのTCP Session)から構成され、複数の最短経路候補を有する

# 提案方式 PD-OLSR

- OLSRの基本部分はそのまま
- TCP通信とUDP通信の特性を考慮した経路生成を行う
  - TCP通信用とUDP通信用のRTを生成
- 各ノードで求めた経路選択指標をOLSRの制御メッセージに乗せて隣接ノードへ広告
- 各ノードは受信した制御メッセージを基に新たに定義した経路計算テーブル(RCT:Route Calculation Table)を生成
  - RCT は宛先ノード(Dest), 宛先への次ホップ, ノード(Next), ホップ数(hop), 通信状態情報(Next の UDP Traffic, Next の TCP Session)から構成され, 複数の最短経路候補を有する

# TCP・UDPの特性 詳細

- UDP
  - 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
  - 輻輳制御によって順調にACKが返ってこればウィンドウサイズを拡大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、ネットワークが更に輻輳することを防ぐ
- つまり
  - TCPとUDP通信が混在するネットワークのトラフィックは
    - ① 送出されるUDPパケットの合計より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
    - ② UDPのパケットロスはそのまま消滅するが、TCPは再送制御を行いながらスループットが最大になるように輻輳制御が働く

TCP効率化は、TCPの輻輳制御がうまく機能するかどうかによって決まる

# TCP・UDPの特性

- UDP
  - 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
  - 輻輳制御によって順調にACKが返ってこればウィンドウサイズを拡大し帯域を使い切ろうとする
- つまり
  - TCPとUDP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

- 経路選択指標

- UDP Traffic

- UDP Traffic とは自身が検出するネットワーク上のキャリアの総量

- TCP Session

- キャリアととして検出する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通信用の経路

- 単純にUDP Trafficが最小の経路を選ぶ

- TCP通信用の経路

- 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

# 提案方式 - 着目点 -

- 着目点

- 現在, インターネット上を流れるトラフィックの9割以上はTCPによるものである[1]
- 今後, UDPによるVoIPや動画のストリーミング配信などの需要がさらに高まる



MANET においても同様な状況が起こりうると

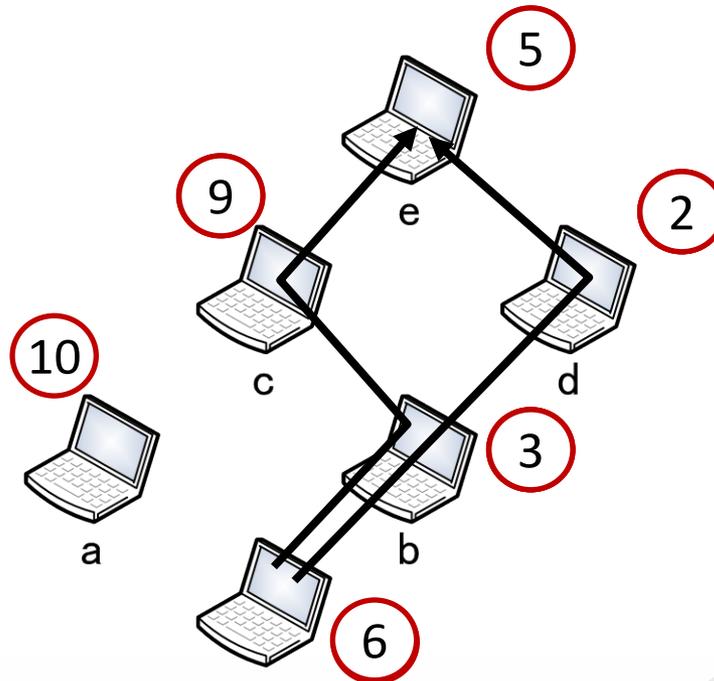
想定

TCPとUDPの混在環境において, 特徴が全く異なる2種類のトラフィックに着目

[1] Thompson, K., Miller, G. and Wilder, R.:Wide-Area Internet Traffic Patterns and Characteristics, IEEE/ACM Trans. Networking,Vol.11, No.6, pp.10-23 (1997)

# 経路全体のトラヒックの場合

- 経路全体のトラヒックを経路選択の指標にすると
  - 通信によって経路のトラヒックが増え, 更新時に経路が頻繁に切り替わってしまう
  - トラヒックの高いノードが含まれる経路ほど振動しやすくなる



- 次ホップノードのトラヒックで考えると経路の振動が起こりにくくなる

# 拡張OLSRの実現方法 ①

## • 情報リポジトリ

**HELLO** , **TC**メッセージを送受信することにより構築

- リンク集合
  - 直接電波の届くノードの集合
- 隣接ノード集合
  - 隣接ノードのアドレスやそのノードの再送信の積極度から成る集合
- 2ホップ隣接ノード集合
  - 隣接ノード集合のさらに1ホップ先に存在するノードの集合
- MPR集合
  - MPRとして選択された隣接ノードの集合
- MPRセレクト集合
  - 自身をMPRとして選択しているノードの集合
- トポロジー集合
  - 3ホップ以上のノードとその隣接ノードを含むネットワークトポロジーの集合

# OLSRの拡張方法②

①UDPトラヒックTCPセッション数をHELLOとTCに付加

③各ノードのUDPトラヒックTCPセッション数リポジトリ内で保持

②各ノードのUDPトラヒックTCPセッション数を収集

④RCTで生成された複数の最短経路の中からUDP・TCP用の経路を選択し、RTを生成

Local Node

UDPトラヒック  
TCPセッション数

情報リポジトリ

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セクタ集合

トポロジー集合

INPUT

HELLO

TC

OUTPUT

Generation

HELLO

TC

Forwarding

TC

RCT

UDPトラヒック, TCPセッション数追加部

宛先ノードへの複数の最短経路の  
UDPトラヒック, TCPセッション数を計算した  
経路計算テーブル(RCT: Route Calculation Table)