

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

三嶋勇太[†] 旭健作[†] 渡邊晃[†]

有線で接続されていたアクセスポイント(AP)間を，アドホックネットワークによって接続する無線メッシュネットワークの研究が注目されている．無線メッシュネットワークのルーティングプロトコルはアドホックルーティングプロトコルや，それを改造したものが使用される．多くのアドホックルーティングプロトコルは，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それぞれの特性を生か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．

Proposal of an Ad-hoc Routing Protocol which takes the Communication Status into Account, and Studies on Lengthy Routing

YUTA MIKAMO[†] KENSAKU ASAHI[†]
AKIRA WATANABE[†]

Studies on the wireless mesh network that connects access points (AP) by an ad-hoc network have been drawing much attention. As the routing protocol for the wireless mesh network, ad-hoc routing protocol or its improved version is used. In many of such ad-hoc routing protocols, it depends on the implementation which route to be chosen when there exist a number of shortest routes. In this paper, we propose an ad-hoc routing protocol that can make route selection that takes advantage of features of both TCP and UDP, by extending the Optimized Link State Routing(OLSR).

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(Access Point)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．また，インフラストラクチャモードにおいて有線接続されていた AP 間を，アドホックネットワークで接続する無線メッシュネットワークの研究に注目が集まっている．

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(以下 RT と記述)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．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．

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．しかし，これらの性質の違いを考慮した経路選択方式は存在しない．そこで，本論文では，アド

[†]名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University

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 の概要について、3 章では PD-OLSR の概要、4 章で経路生成方法を示し、5 章でシミュレータ上での実装について示す。6 章で冗長経路についての検討、7 章でまとめを行う。

2. OLSR

2.1 OLSR の概要

OLSR(Optimized Link State Routing)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 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 生成プロセスを示す。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する。図 1(b)に示す RT は、ノード b を対象としてノード m および n 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 m および n へのそれぞれ 1 つの最短経路が生成される。図 1 ではノード b からノード m への最短経路[b→d→g→k→m]が生成された様子を示している。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、ノード g から f への通信がすでに行われていたものとする。ノード g から送信されるキャリアは隣接ノード c, d, f, h, j, k

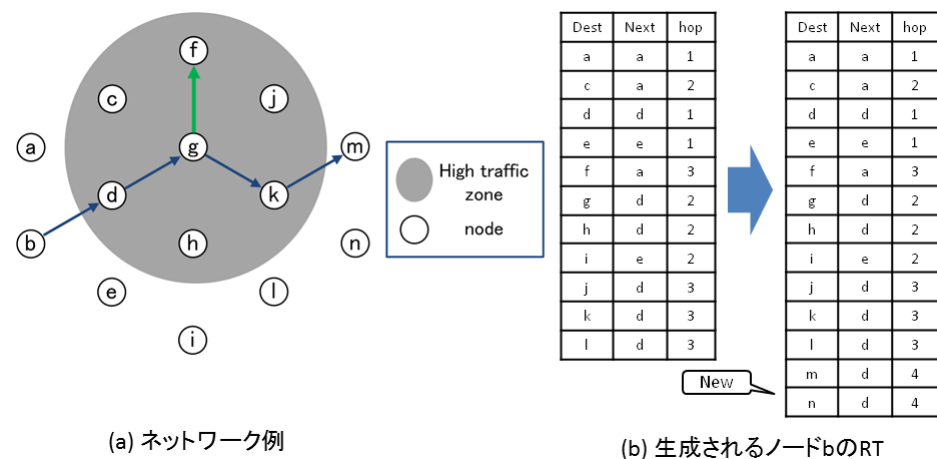


図 1 OLSR の RT 生成

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 g 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。

3. PD-OLSR の概要

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 が順調に返ってくるとウィンドウサイズを拡大し、帯域を有効に使用しようとする。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。

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ま利用し、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 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するネットワーク上の受信可能な電波(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。また、キャリアには、UDP および TCP の両方が含まれる。

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、カーナビゲーションシステムの経路探索や、鉄道の経路案内においても用いられている。

既存の OLSR では、ネットワークトポロジの情報からホップ数を基準に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、通信状態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などが可能である。

4. PD-OLSR の経路生成手法

4.1 生成される経路例

図 2 に PD-OLSR で生成される経路例を示す。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 2(a)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 g からノード f への通信が行われているため、隣接ノードであるノード c, d, h, j, k 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。

4.2 経路探索用テーブル RMT

PD-OLSR においては RMT(Route Metric Table)と呼ぶ経路計算用テーブルを新たに定義する。図 3 にノード b における RMT と RT の例を示す。経路中のトラフィックの合

計をコストと呼ぶこととする。RMT は宛先ノード(Dest)、宛先へ経路のコスト(Cost)、経路のホップ数(hop)、経由するノード(hop1,hop2,...)の情報から構成される。

4.3 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。経路探索結果を、RMT に保存する。ここで、リンクメトリックへの変換は、両端ノードの UDP traffic の和を経路コストをリンクメトリックとすることで行う。

PD-OLSR の RT は探索された最適経路の各宛先について、RMT の Dest, hop1, hop を RT における Dest, Next, hop として保存する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 2 でノード b からノード m 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[b→e→i→l→n→m]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演算することで生成できる。TCP 用の RT 生成の場合、各 Dest に対して最小 TCP Session 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。

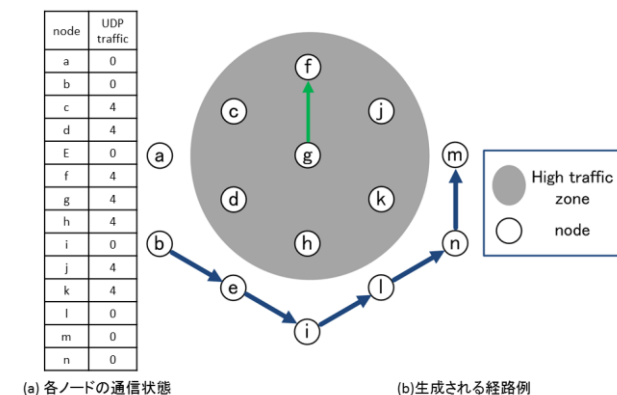


図 2 PD-OLSR で生成される経路例

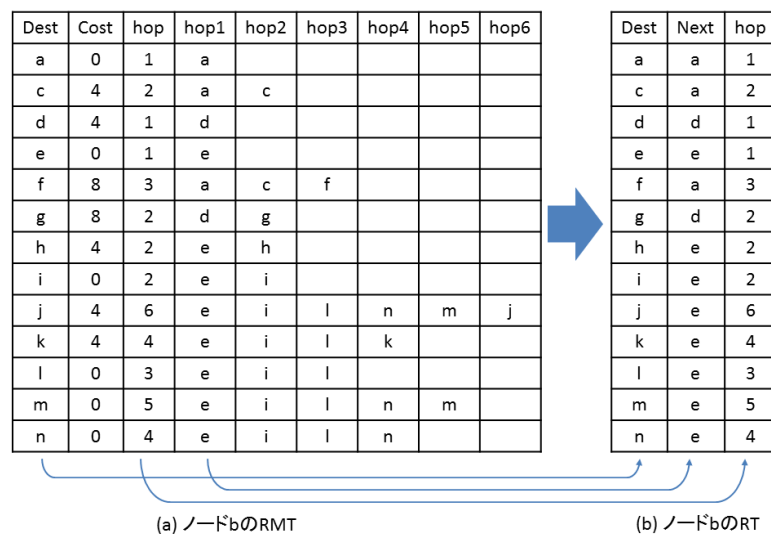


図 3 PD-OLSR の RT 生成

5. シミュレータへの実装

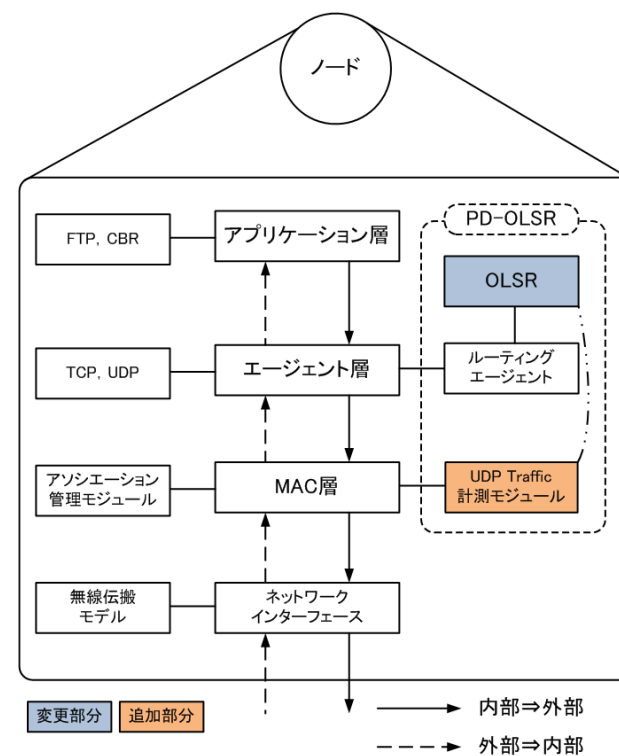
PD-OLSR をネットワークシミュレータ ns-2[12]実装する方法を検討した．UDP 用の RT 生成機能を例にして，以下にその概要を示す．

5.1 ns-2 の変更部分

図 4 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．

5.2 OLSR の拡張方法

OLSR において，制御メッセージと情報リポジトリには図 5 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．OLSR の改造において，図 5 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．また，情報リポジトリ内に新たにダイクストラ法による経路探索結果を保存する RMT を追加した．

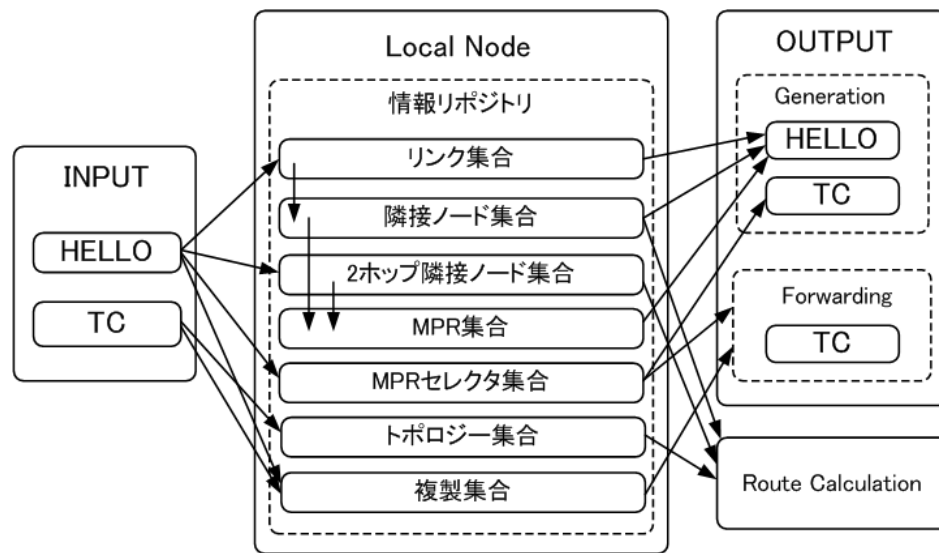


図 5 制御メッセージと情報リポジトリの関係

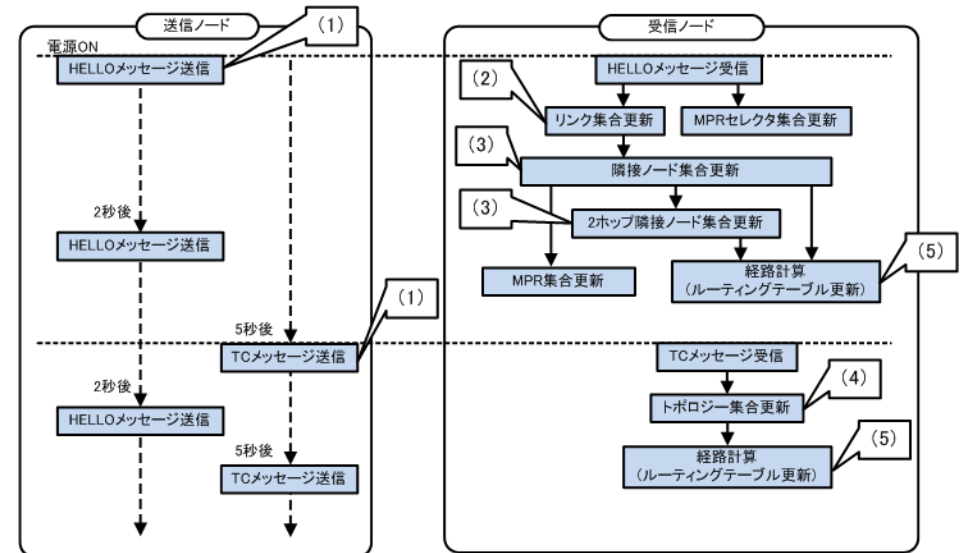


図 6 OLSR の改造箇所

5.3 制御メッセージの処理

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 6 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生成
- (5) 経路計算
 - 経路計算(RT 更新)プロセスに先立ち、4 章に示す方法で RMT を生成
 - RMT から UDP 用 RT を生成

6. 冗長経路に係る検討

6.1 冗長経路に起因するスループット低下

PD-OLSR での経路探索では、通信状態を測定し、トラフィックの少ないノードを中継するように経路を選択する。その際、図 7 に示すように、ホップ数を過大に増やし、トラフィックの高い部分を避ける経路が選択される可能性がある。

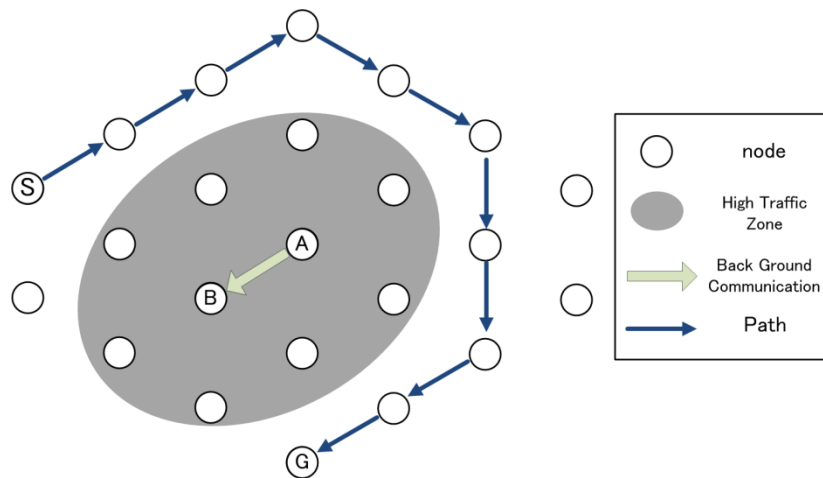


図 7 hop 数を過大に増加させた経路例

アドホックネットワークでは、遠隔のノードとはマルチホップ通信を行う。通信によって中継ノードの数、つまりホップ数が異なり、ホップ数が増えていくと送信元から宛先までのパケット到達率は低下する。

そこで、ホップ数を経路コストに取り入れることでホップ数が過剰に増える経路選択を防止することが望ましい。

6.2 ホップ数を考慮した経路コスト計算

経路生成時の経路コストを、式(1)に示すような計算式によって求める。このとき、最短経路のホップ数を h 、リンクメトリックの合計を P 、経路コストにおけるホップ数をどのように考慮するのかを決定するホップ数 h の関数 $f(h)$ 、求めるホップ数を考慮した経路コストを C とする。

$$C = P + f(h) \quad (1)$$

式(1)中の $f(h)$ は、経路のホップ数増加によるスループットの低下および、中継ノードのトラフィック増加によるスループット低下の関係を検証し、どのような形を用いるか決定する。

7. むすび

OLSR を拡張することにより、TCP 用と UDP 用に別々の RT を生成し、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。RT を分けることで、TCP と UDP で通信の特性が異なるという点を経路探索に反映することができる。さらに、ダイクストラ法を用いることで、最短経路に限らず、高トラフィックのノードを回避した最適な経路選択が可能となる。

今後は、PD-OLSR の実装を完了し、UDP、TCP、および UDP と TCP が混在する環境においてのシミュレーションを行い、PD-OLSR の効果を確認する。また、経路コストを計算する際に、ホップ数をどのようにコストに反映するかを検討する。

参 考 文 献

- [1] T. Clausen, Ed. : “Optimized Link State Routing Protocol (OLSR)”, RFC 3626 (2003)
- [2] D. Johnson : “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4”, RFC 4728 (2007)
- [3] C. Perkins : “Ad hoc On-Demand Distance Vector (AODV) Routing”, RFC 3561 (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: Temporally-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 SIGCOMM Computer Communication Review, pp. 83–88 (2003).
- [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, 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).
- [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/>.



通信状態を考慮した経路選択を可能にする 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

名城大学大学院 理工学研究科
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など無線LANを利用可能な端末の普及に伴い、アドホックネットワークに関する研究が注目されている
- ▶ アドホックネットワー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
 - Access Point同士がアドホックモードで接続

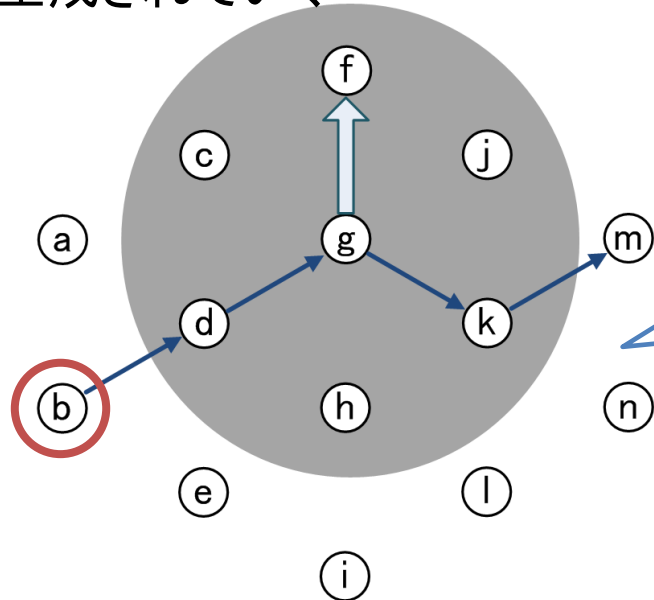
OLSRの概要 (Optimized Link State Routing)

- ▶ 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

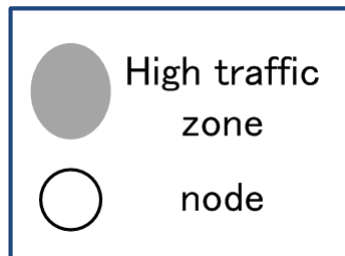
制御メッセージには
リンク情報のみ

OLSRのRT

- 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



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

ノードbのRT

Dest	Next	hop
a	a	1

ノードbのRT

Dest	Next	hop
a	a	1

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 → f

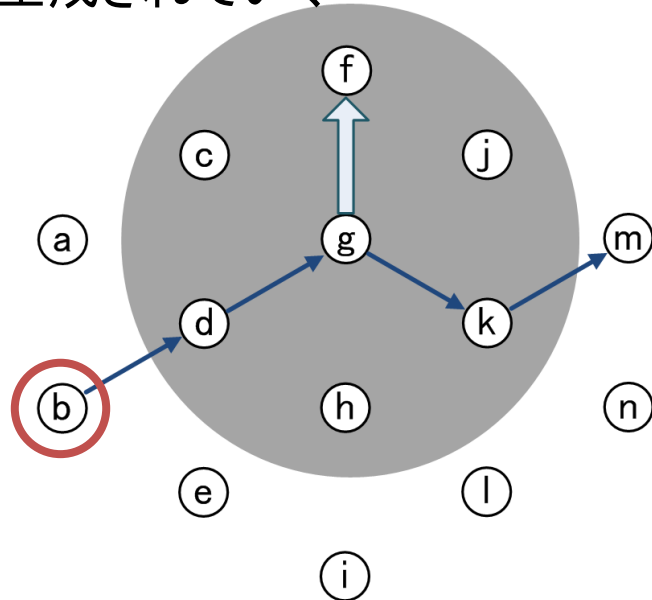
g	d	2
h	d	2
i	e	2
j	d	3
k	d	3
l	d	3

g	d	2
h	d	2
i	e	2
j	d	3
k	d	3
l	d	3
m	d	4
n	d	4

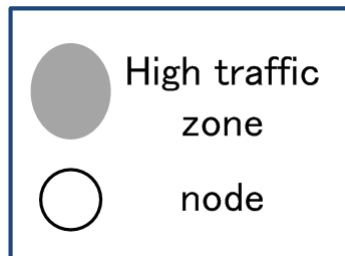
New

OLSRのRT

- 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



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

ノードbのRT

Dest	Next	hop
a	a	1
c	a	2
d	d	1
e	e	1
f	a	3
g	d	2
h	d	2
i	e	2
j	d	3
k	d	3
l	d	3

ノードbのRT

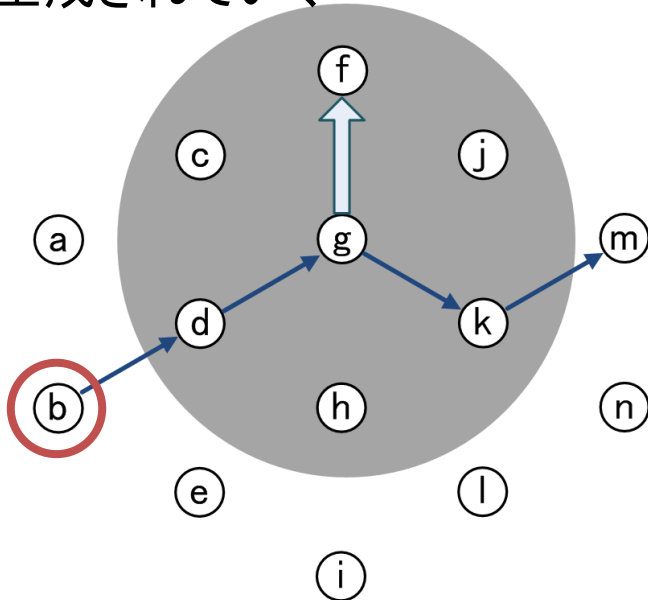
Dest	Next	hop
a	a	1
c	a	2
d	d	1
e	e	1
f	a	3
g	d	2
h	d	2
i	e	2
j	d	3
k	d	3
l	d	3
m	d	4
n	d	4

New

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

ノードbのRT

Dest	Next	hop
a	a	1
c	a	2
d	d	1
e	e	1
f	a	3
g	d	2
h	d	2
i	e	2

ノードbのRT

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

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定義されていない

○ node

New

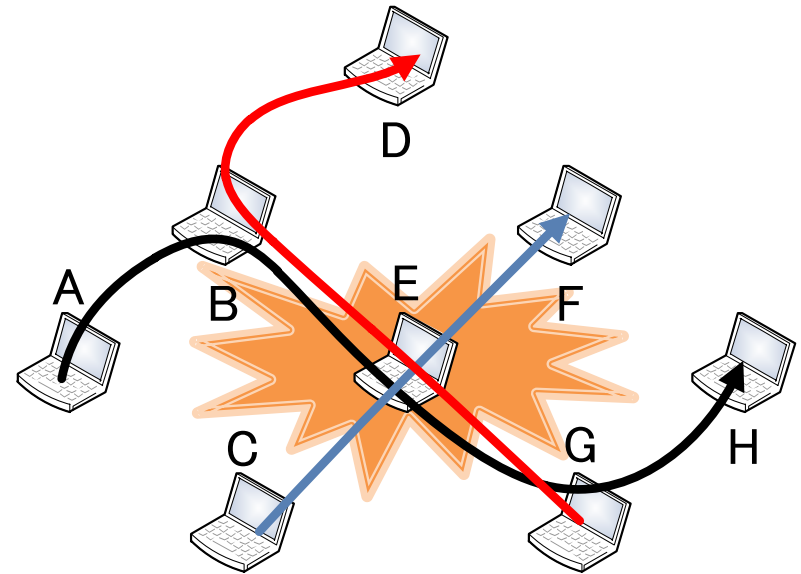
課題1

- ▶ 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



TCP・UDP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

課題2

- ▶ TCP通信ではパケットロスが発生するとウィンドウサイズを縮小する
- ▶ UDP通信ではTCP通信のような輻輳制御を行わない
- ▶ 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を使用する



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

提案方式:PD-OLSR

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

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

PD-OLSR トラフィック情報

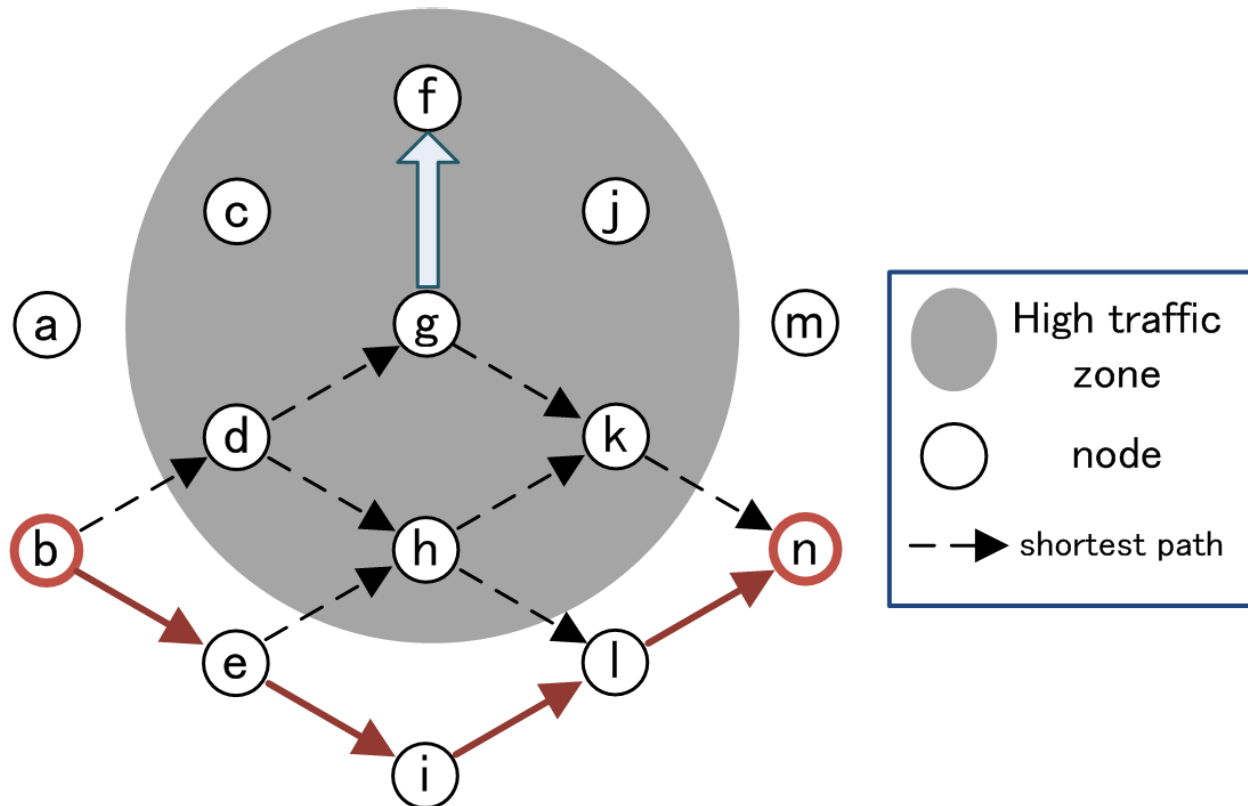
- ▶ トラフィックの高いノードを避けるための経路選択指標
- ▶ UDP通信:UDP Traffic
 - 自身が検出するネットワーク上のキャリアの総量
- ▶ TCP通信:TCP Session
 - 自身が検出するTCPセッション数の合計
 - 各セッションがウィンドウサイズを増減
 - セッション間の公平を保つため

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

経路探索

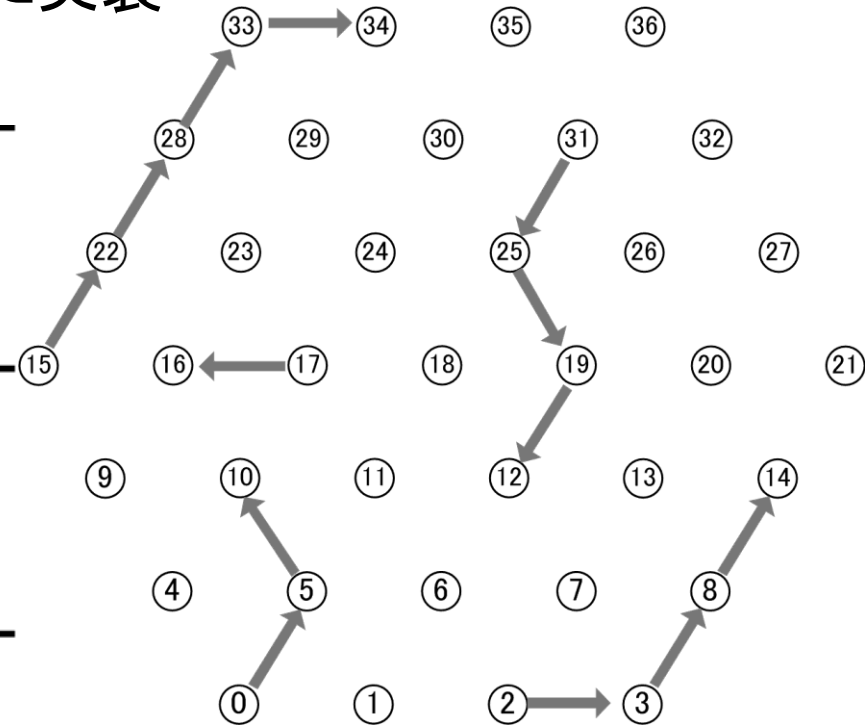
- ▶ 最短経路が複数存在する場合
→ トラフィックの低い経路を選択



シミュレーション

▶ UDP通信用RT生成機能をns-2に実装

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] (1hop) 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]



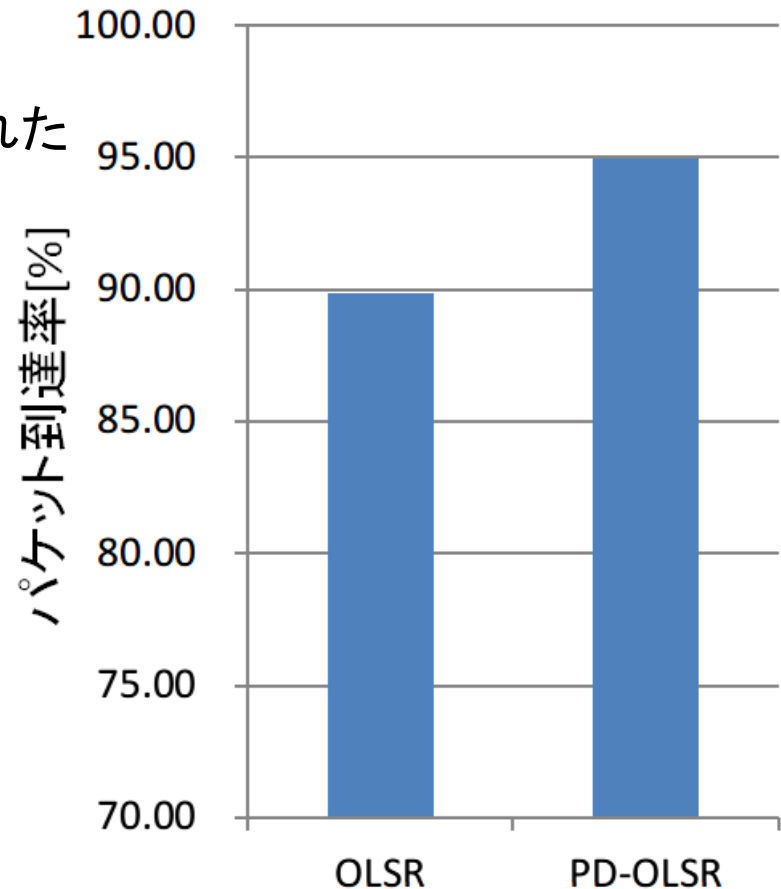
開始30秒後からUDP通信を10秒間隔ごとに増加
計530秒間OLSRとPD-OLSRの場合で10回ずつ
ネットワーク全体のパケット到達率を比較

評価・拡張

- ▶ パケット到達率が平均約6%改善
 - 10回のシミュレーション全てで効果が見られた
- ▶ 最短経路の中から最適な経路選択
→ 最短経路の中に
最適な経路があるとは限らない

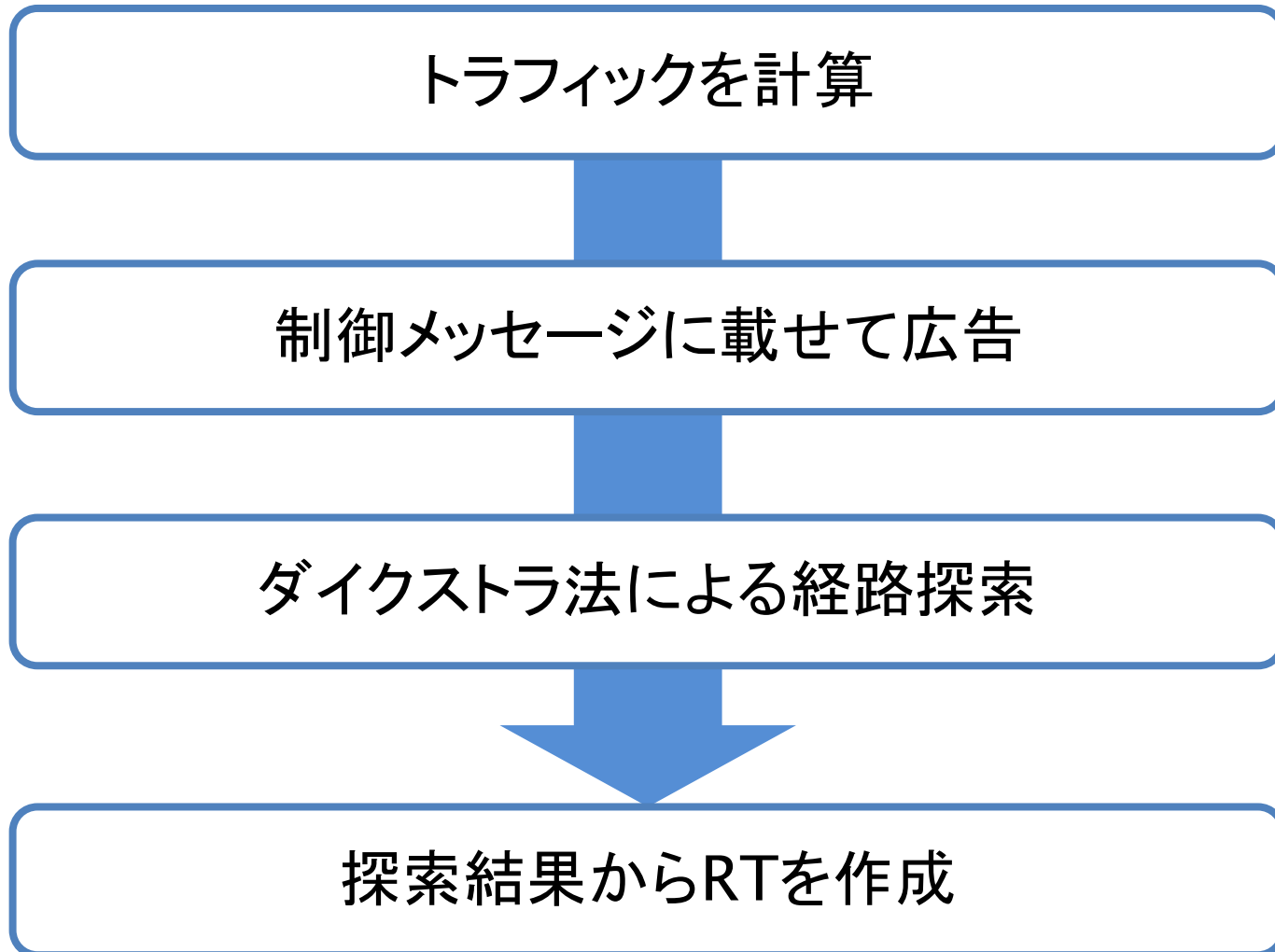


- ▶ 最短経路に限らず最適な経路を選択
 - 経路探索にダイクストラ法を用いる



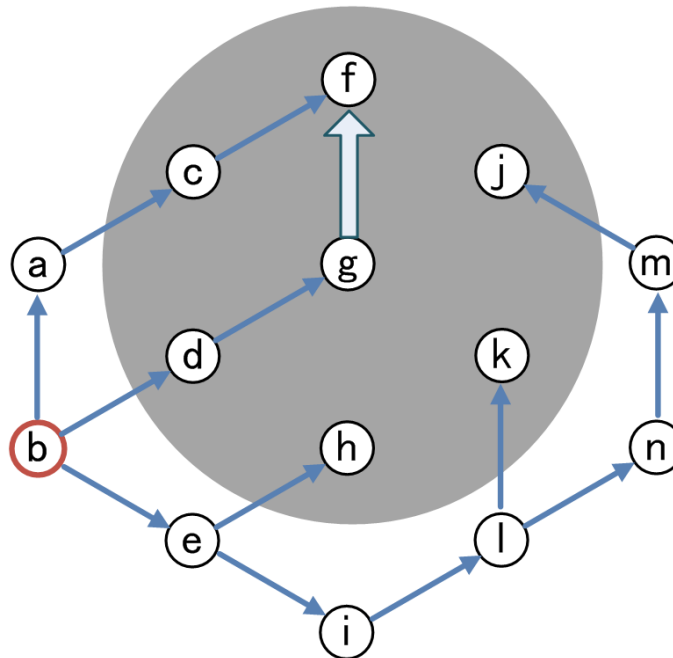
冗長経路選択を可能にする拡張

処理の流れ



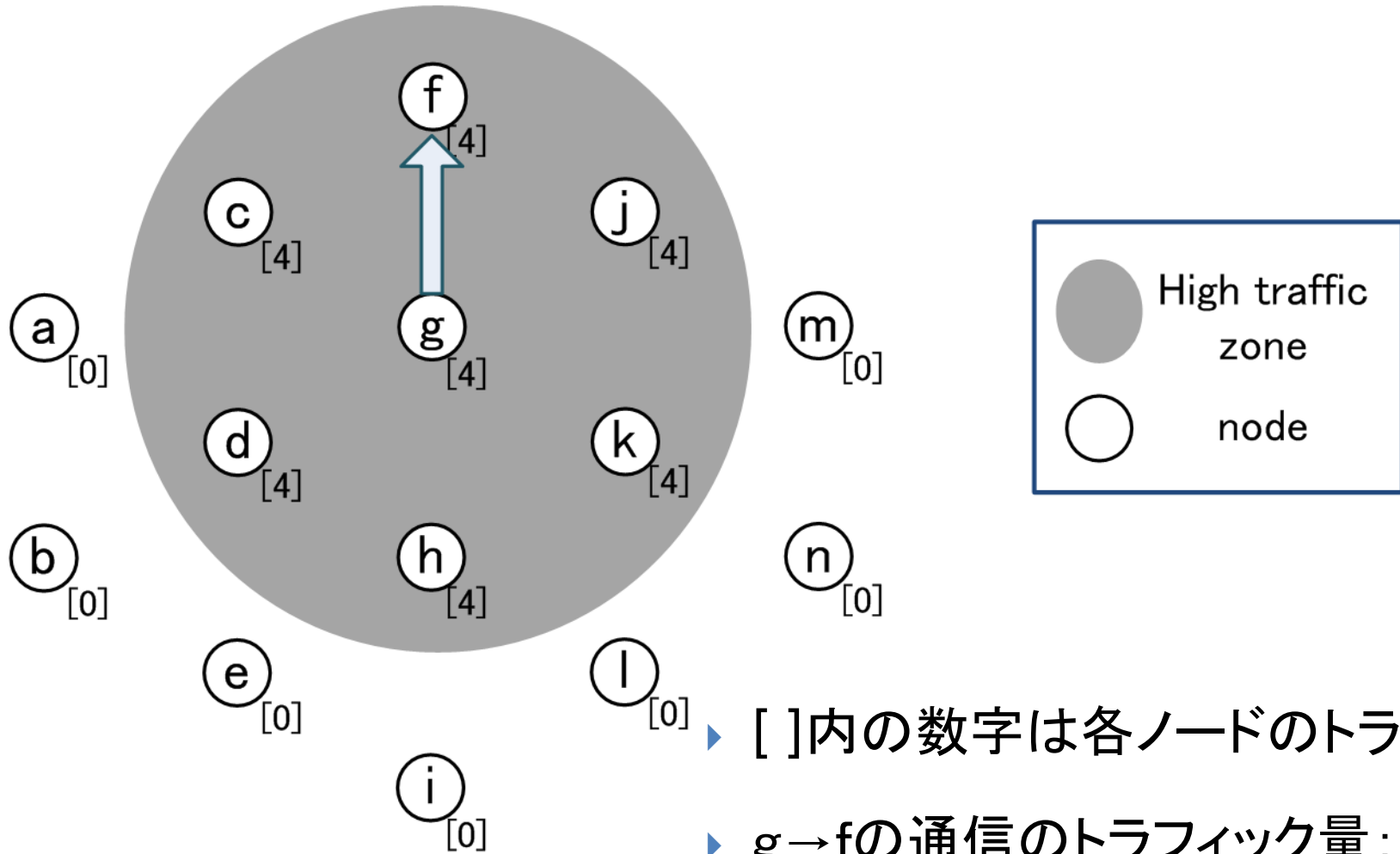
経路探索

- ▶ ノードごとのトラフィック情報からリンクメトリックに変換
 - リンクメトリック: リンクの両端ノードのトラフィック情報の和
- ▶ ダイクストラ法を用いた経路探索
 - ダイクストラ法: グラフ理論による最短経路問題解決アルゴリズム
 - 各宛先ノードに対して経路コスト, 経路中の中継ノード, ホップ数 that 得られる



動作

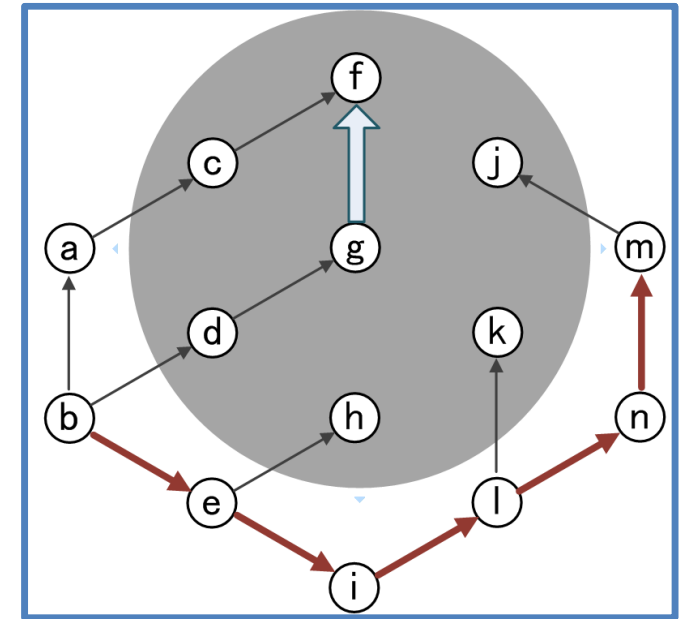
動作1



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

動作2

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		

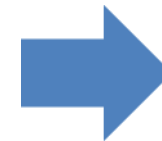


Dest :宛先
 Cost :経路コスト
 hop :ホップ数
 hop1,hop2,...:中継ノード

動作2

ノードbのRT

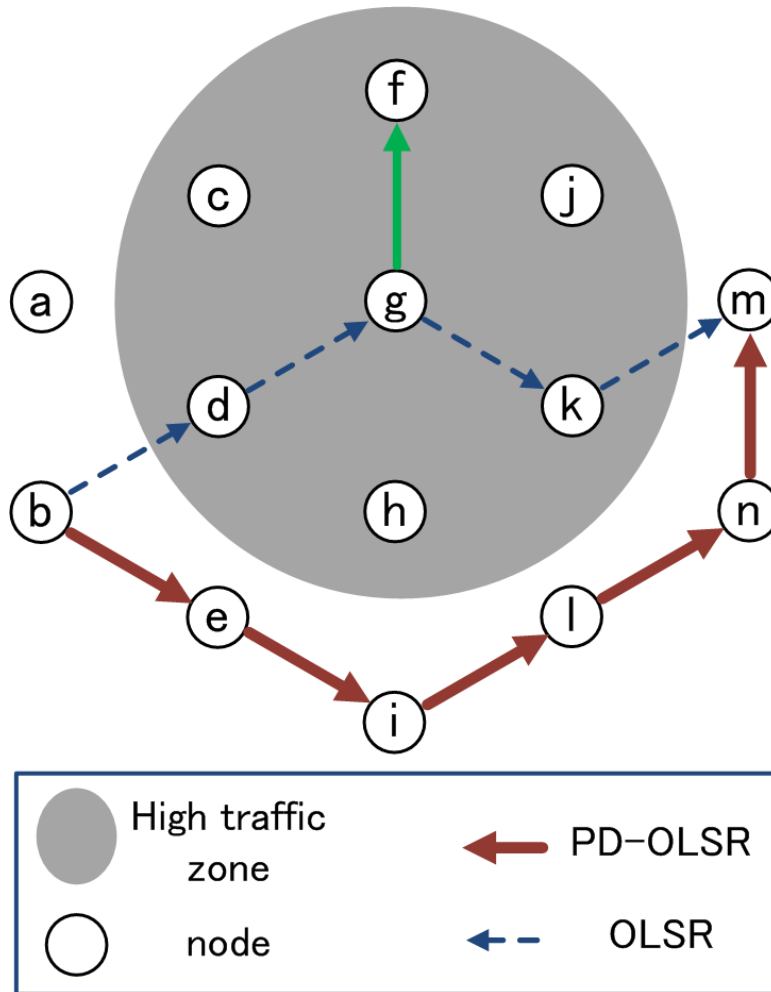
Dest→Dest
hop→hop
hop1→Next
としてRTに保存



Dest	hop	hop1
a	1	a
c	2	a
d	1	d
e	1	e
f	3	a
g	2	d
h	2	e
i	2	e
j	6	e
k	4	e
l	3	e
m	5	e
n	4	e

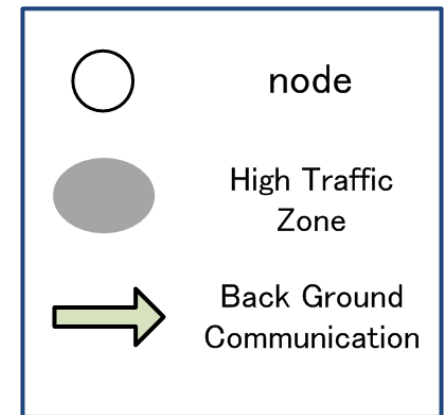
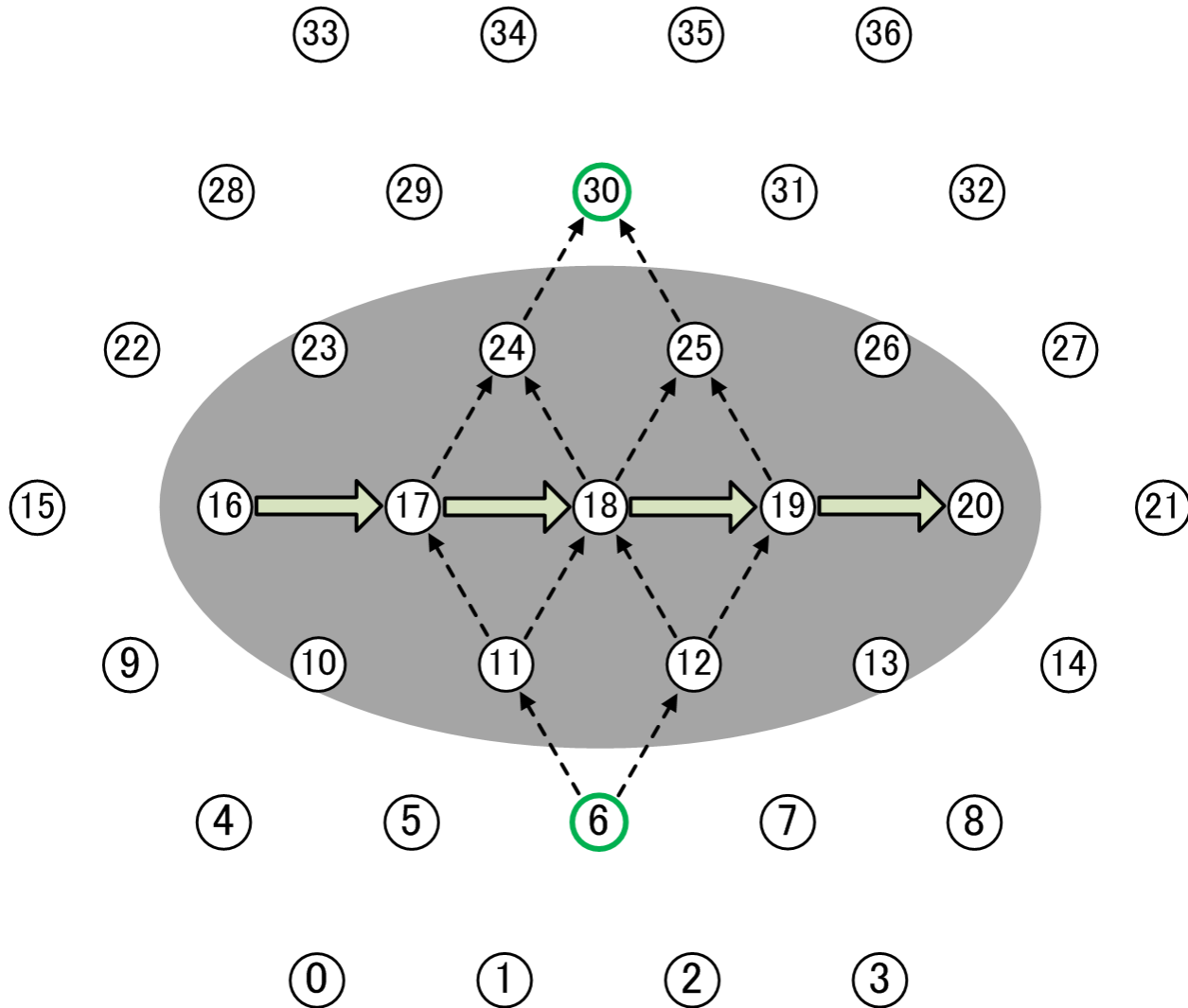
Dest	Next	hop
a	a	1
c	a	2
d	d	1
e	e	1
f	a	3
g	d	2
h	e	2
i	e	2
j	e	6
k	e	4
l	e	3
m	e	5
n	e	4

経路比較

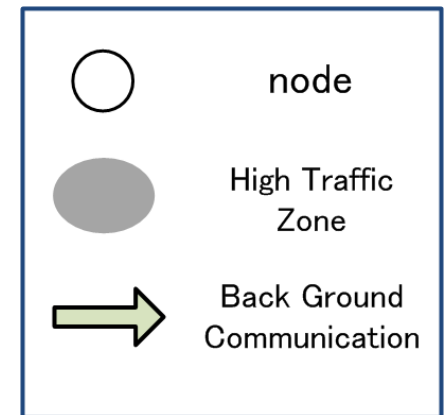
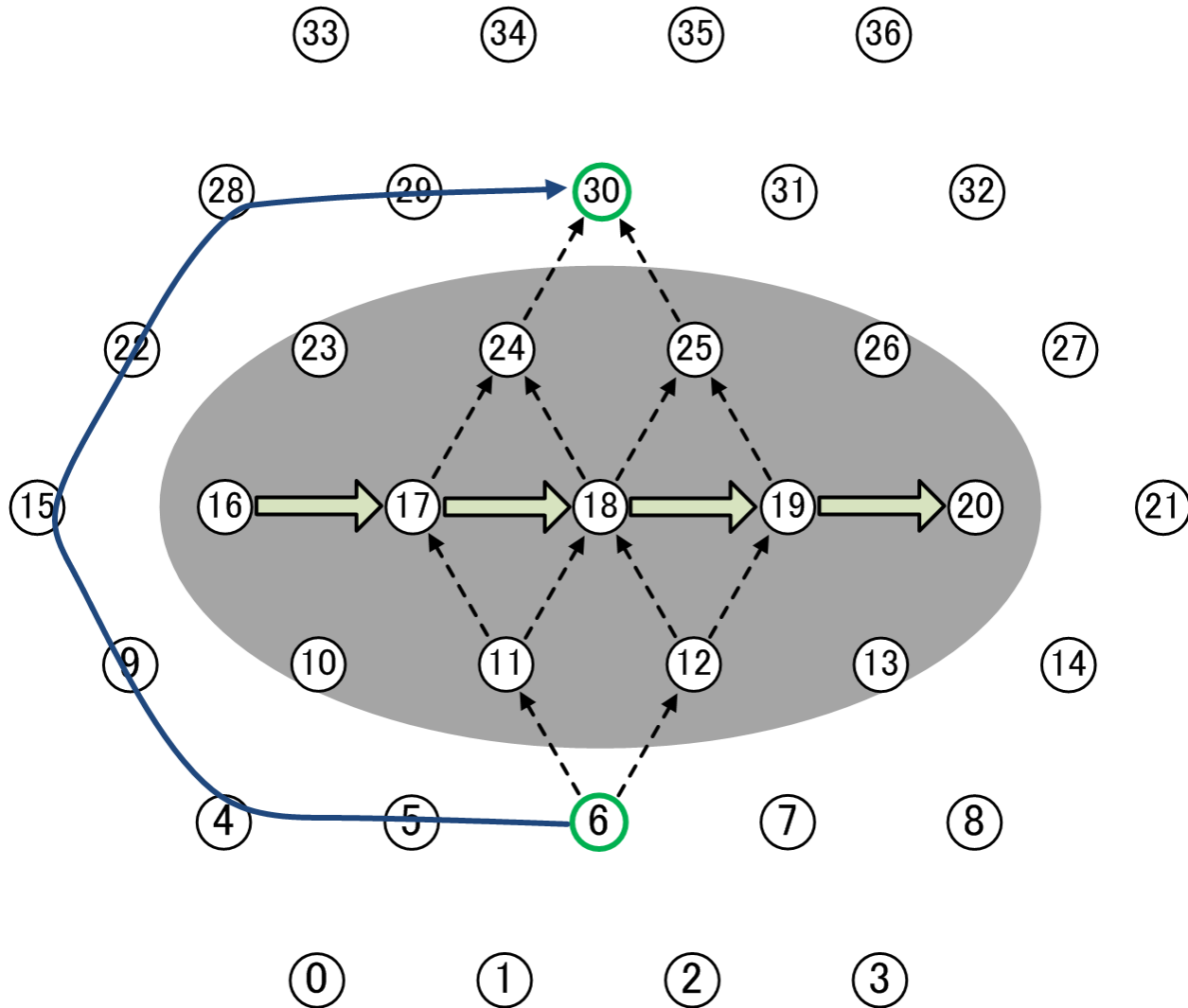


- ▶ トラフィックの高いノードを避けた経路を生成
- ▶ 経路コストがより小さければホップ数を増やした経路を選択可能
 - 例での各経路のホップ数
 - OLSR : 4hop
 - PD-OLSR : 5hop

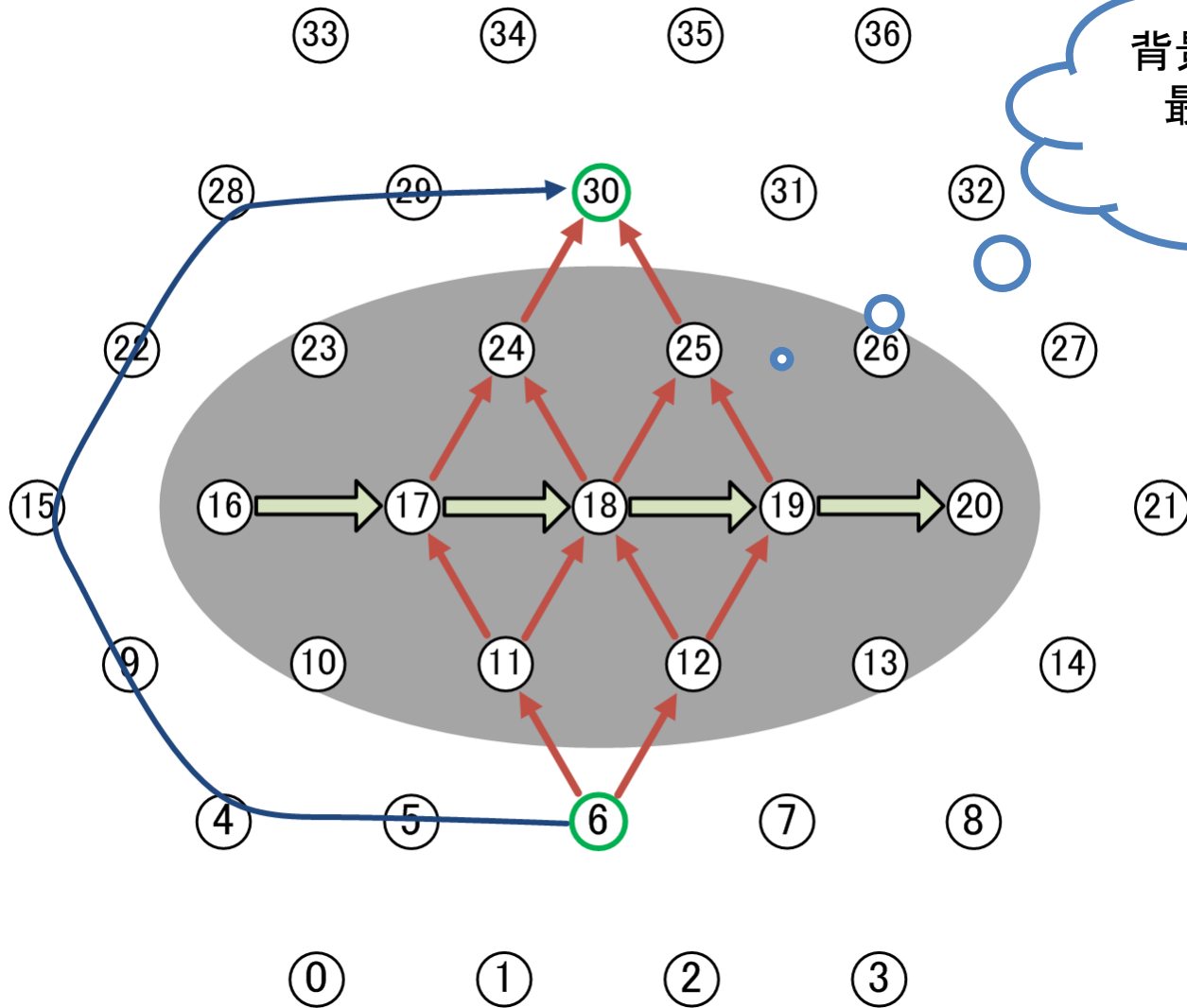
迂回経路



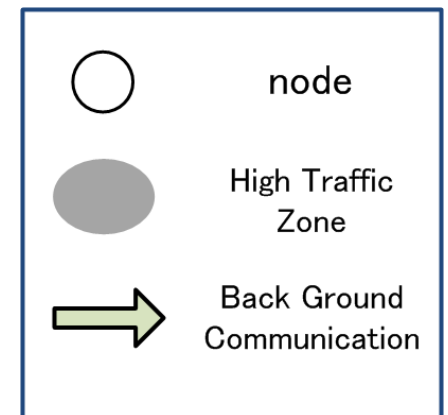
迂回経路



迂回経路



背景負荷によっては
最短経路の方が
いいかも？



ホップ数を考慮した経路コスト

- ▶ 現在の提案方式: 通信状態のみを基準に経路探索
- ▶ どれほど遠回りであっても通信状態指標の低いノードを経由する経路が選択される
- ▶ アドホックネットワークではホップ数が増えることによってパケット到達率が低下する
- ▶ ホップ数を考慮した経路コスト

$$C = P + f(h)$$

C : 経路コスト

P : 経路中のノードの通信状態指標の合計

h : ホップ数

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
- 複数の最短経路の中から最適な経路を選択する方式では平均6%のパケット到達率改善がみられた
- 最短経路に限らず最適な経路を選択できる拡張を提案した

▶ 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
- ホップ数を経路選択指標にどのように反映させるのか検討

