

無線アドホック網のルーティングプロトコル OLSRにおける経路表作成時間の数理解析手法

川西 俊之^{†1} 紀 一 誠^{†2}

無線アドホックネットワークのルーティングプロトコルである OLSR の経路表作成時間を時間的に非斉時なマルコフ連鎖の吸収時間として求める数理解析法を示す。OLSR は端末の網内移動が比較的少ないネットワークで用いられ、データ通信を開始する前に経路情報を確立する点に特徴がある。OLSR は隣接ノード間の接続情報のみを知らせあう Hello メッセージと、遠隔ノード間の接続情報を知らせる TC メッセージを用いて経路表を作成する。この 2 種類のメッセージは異なる周期を持つ一定時間間隔で送信される。本論では、経路表完成状態を吸収状態に持つマルコフ連鎖の吸収時間として、各端末が情報を持たない状態から経路表が完成するまでの時間を求める。2 種類のメッセージの送信時点や情報破棄時点ではシステムの状態推移構造が変わるため、時間的に非斉時なマルコフ連鎖を用いる。

Analytical Method for Routing Table Calculation Time of Ad-hoc Network Routing Protocol OLSR

TOSHIYUKI KAWANISHI^{†1} and ISSEI KINO^{†2}

The OLSR is a routing protocol for Mobile Ad-hoc Networks. The protocol creates routing table in the network layer. We propose a Markovian stochastic model for performance evaluation of the OLSR in this paper. The routing table calculation time in the network layer is used as performance measure. We limit the discussion to linear network topology for the simplification. Numeric examples are shown in some scenarios.

^{†1} 株式会社情報工房
Johokobo Inc.

^{†2} 神奈川大学理学部
Faculty of Science, Kanagawa University

1. はじめに

近年、情報ネットワークの通信プロトコルが見直されている。それは無線ネットワーク技術の進展によるものである。無線ネットワークは、従来の有線ネットワークにはなかった特有の問題を有している。たとえば、ノードが移動することによる環境の変化、無線帯域の制限、電源の制限などである。特に、無線ネットワークの中でも無線アドホックネットワークと呼ばれる利用形態ではこの特徴が顕著にあらわれる³⁾。無線アドホックネットワークとは、アクセスポイントなどの特定のインフラストラクチャを持たずに、ネットワークを構成するノードが自律分散的に通信を行うようなネットワークの利用形態のことである。特にノードが移動するような無線アドホックネットワークのことをモバイルアドホックネットワーク (MANET) という。

そのような背景があり、MANET 用に様々な通信プロトコルが提案されている。特に、ノード間の通信経路を確定するためのルーティングプロトコルがインターネットの標準化団体である Internet Engineering Task Force (IETF) で議論されている。IETF では MANET 用のルーティングプロトコルを「プロアクティブ型」と「リアクティブ型」の 2 種類に分類している。両者は経路情報を交換するタイミングが異なる。プロアクティブ型プロトコルは、データ通信を行う前に経路情報を交換することで経路を構築する。その後、定期的な情報を交換することで経路をメンテナンスする。それに対して、リアクティブ型プロトコルは、データ送信時に経路情報を交換する。毎回のデータ送信時にオンデマンドで経路を作成するため、経路作成のオーバーヘッドがかかるが定期的な情報交換は不要であるという特徴を持つ。

Optimized Link State Routing Protocol (OLSR)²⁾ は代表的な MANET 用のプロアクティブ型ルーティングプロトコルである。このプロトコルはインターネットの標準化文書である Request For Comments (RFC) 3626 で規定されている。OLSR は経路を構築するための情報を Hello メッセージと TC メッセージという 2 種類のメッセージに格納し交換する。

OLSR などのルーティングプロトコルの性能評価にあたっては、ネットワークシミュレーションツールを用いた評価が一般的であり、いくつかの結果が報告されている^{1),5)}。しかしながら、数理的に解析するためのモデルは提案されていない。そこで本論では、OLSR の性能を評価するための数理モデルを提案する。具体的には、OLSR が経路を確定するまでにかかる時間を時間的に非斉時な吸収マルコフ連鎖を用いて評価する。

本論の構成は以下のようになっている。次章では、OLSR の概要についてまとめる。ここでは、OLSR の経路作成過程とメッセージの交換シーケンスについて重点的に述べる。その後、3 章で数理モデルを提案する。4 章で解析例を示し、5 章で本論をまとめる。

2. OLSR の機能

本章では、OLSR の機能について説明する。まず、OLSR を含んだネットワーク層のプロトコルの概要について述べ、その後、OLSR の機能について説明する。

2.1 ネットワーク層のプロトコルの役割

通信ネットワークのプロトコルは階層的に管理されており、各階層が役割分担をして最終的な通信の目的を達成する。今回対象とする OLSR はこの階層のうち、ネットワーク層で動作する。このネットワーク層の主要なプロトコルにインターネットプロトコル (IP) がある。IP は、送信元の端末から宛先の端末へデータを届ける役割を負っている。つまり、IP が正常に機能すれば、端末間が物理的に直接通信できない場合でも、他の端末をうまく中継させることでデータを配送することができる。この機能を果たすためには、ある宛先にデータを届けるためにどの端末にデータを中継させればよいかという情報が必要となる。この中継情報を収集するのがルーティングプロトコルである。ルーティングプロトコルは情報収集用のメッセージを交換するなどの方法で、中継用の経路情報を収集し、経路表と呼ばれるテーブルに格納する。ルーティングプロトコルが作成した経路表をもとに、IP がデータを端末から端末へと伝送する。よって、ネットワーク層での性能は、IP の性能はもとより、ルーティングプロトコルの性能に依存するといえる。

2.2 OLSR プロトコルの概要

OLSR は、代表的な MANET 用のプロアクティブ型ルーティングプロトコルである。通信ネットワークのプロトコル階層のネットワーク層で動作するモジュールであり、他のルーティングプロトコルと同様、経路表を作成することを目的とする。OLSR の動作を定めた RFC 3626 には、OLSR の主要機能として以下の 3 つの項目があげられている。

- (1) リンク感知および隣接ノード探知：OLSR は、隣接ノードからメッセージを受信することで隣接ノードとの間の無線リンクを発見する。この機能のことを「リンク感知」と呼ぶ。さらにメッセージの交換を続けると、隣接ノードは互いに、他方から送信されたメッセージを受信することができる。そうすることで両者はお互いに双方向で到達可能であることを知ることができる。このような両者の間の無線リンクを対称リンクと呼ぶ。また対称リンクでつながったノードの関係を対称隣接関係と呼ぶ。

この対称隣接ノードを発見する機能が「隣接ノード探知」である。この両機能は、隣接ノード間の関係のみに着目した機能である。

- (2) トポロジ開示：「トポロジ開示」機能は、ネットワーク全体に関わる機能である。ノードの隣接情報をネットワーク中に転送することにより、遠隔ノードがネットワーク構造 (トポロジ) を理解することができる。
- (3) 経路表作成：最終的に、OLSR は隣接ノード情報とトポロジ情報を用いて経路表を作成する。

OLSR は主に Hello メッセージ、トポロジコントロール (TC) メッセージの 2 種類のメッセージを交換する。Hello メッセージはリンクと隣接ノードの情報を伝達する。このメッセージはリンク感知と隣接ノード探知に使用される。そのため、メッセージを生成したノードの隣接ノードのみがこのメッセージを受信する。それに対して、TC メッセージはトポロジ情報を格納し、ネットワークの全ノードに配信される。これらのメッセージによって伝達された情報は、OLSR が動作している各ノードが個別に蓄積していく。

また、TC メッセージを効率的に全ノードへ転送するための仕組みとして、Multipoint Relay (MPR) ノードという選ばれたノードのみにメッセージの転送を任せる仕組みが OLSR には備わっている。全ノードが通信可能となるために必要な最小限のノードのみを MPR として選択し、MPR のみがおノードのメッセージを転送することで、複数ノードが重複してメッセージを転送することを防ぐ。なお、MPR の選択アルゴリズムに関しては、今回のモデル化に直接関連しないので詳細を省略する。

2.3 OLSR のプロトコル規約で定義されている用語

OLSR のプロトコル規約が定義されている RFC 3626 では以下の用語が定義されている。本論でもこれに準じた用語を使用する。

- ノード：OLSR が実装されているルータのことをいう。各ノードがルータとしての機能とともに、クライアントやサーバとしての機能も備えている。
- 隣接ノード：他ノードを経由せずに、無線通信によって直接通信できるノードのことをいう。
- 2 ホップ隣接ノード：1 台の隣接ノードを経由することにより、到達することのできるノードのことをいう。
- MPR：メッセージを全ノードに転送するために最低限必要な中継ノードのことをいう。
- MPR Selector：あるノードを MPR として選択しているノードのことを、そのノードの MPR Selector という。

- 対称リンク：OLSR ノードをつないでいる，双方向の無線リンクのことをいう．
- 非対称リンク：OLSR ノードをつないでいる，片方向の無線リンクのことをいう．
- 対称隣接関係：ある 2 つのノードが対称リンクでつながっていることを，両者は対称隣接関係にあるという．

2.4 情報リポジトリ

OLSR がメッセージ交換によって得た情報は各ノードの情報リポジトリに格納される．情報リポジトリは以下のような構造をしている．

- **Multiple Interface Association Information Base** : Interface Association Set .
- **Local Link Information Base** : Link Set .
- **Neighborhood Information Base** : Neighbor Set , 2-Hop Neighbor Set , MPR Set , MPR Selector Set .
- **Topology Information Base** : Topology Set .
- **Host and Network Association Information Base** : Host and Network Association Set .

上記のリポジトリ内のテーブルのうち，主に Neighborhood Information Base の Neighbor Set , 2-Hop Neighbor Set さらに Topology Information Base の Topology Set が本論での提案手法に関連するため，次節以降で詳細を記す．

2.5 メッセージの遅延

OLSR が生成する Hello および TC メッセージは，それぞれ Hello Interval , TC Interval という一定間隔で送信されるが，無線通信の影響および OLSR の機能により伝達過程で遅延が発生し，メッセージ受信タイミングにずれが生じる．そこで，OLSR の各機能に関するメッセージ交換シーケンスについて述べる前に，無線通信や OLSR の仕組みからくる遅延が OLSR のメッセージ交換に与える影響について説明する．

先に述べたとおり，OLSR はネットワークのプロトコル階層のうち，ネットワーク層で動作するプロトコルである．この下位層として働き，隣接無線ノード間のパケット伝達を管理するのが無線の MAC プロトコルである．代表的な無線の MAC プロトコルとして，無線 LAN として知られている IEEE 802.11⁴⁾ があげられる．IEEE 802.11 には様々な追加仕様が あるが，いずれも Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) という方式でメッセージを交換する．この方式では，メッセージを送信する際に，微小な遅延を発生させる．これにより，無線区間でメッセージが衝突し判別不能となることを防ぐ．なお，この遅延は確率的に決定される．また，メッセージが衝突するなどしてロスした場合

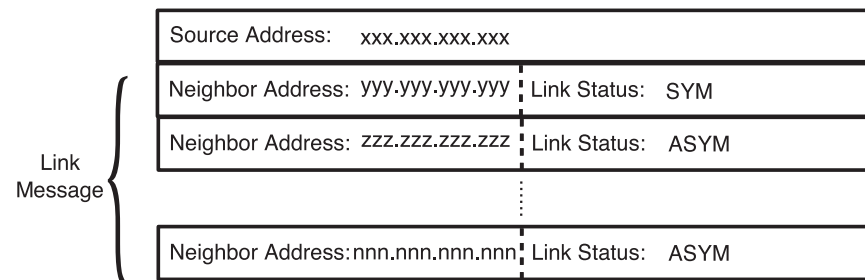


図 1 Hello メッセージの構造
Fig. 1 Structure of Hello Message.

は，一定回数再送する．

また，この MAC プロトコルによる遅延とは別に，メッセージの無線区間での衝突を防ぐため，OLSR にはメッセージ送信の際にジッタと呼ばれる遅延を発生させる機能が独自に備わっている．このジッタも確率的に決められる．具体的には，ジッタの最大値 MAXJITTER を定め，区間 $[0, \text{MAXJITTER}]$ の一様分布に従って実際のジッタを決定する．この MAXJITTER の値は，OLSR の各実装に依存する．ただし，ジッタで送信を遅らせた分だけ実際のメッセージ送信間隔を短くすることにより，メッセージ送信の基点となる時刻が一定間隔となるようになっている．

なお，IEEE 802.11 のような MAC プロトコルによる遅延はマイクロ秒からミリ秒単位のオーダの遅延であり，OLSR のジッタはミリ秒から秒単位のオーダの遅延であるため，メッセージの受信タイミングへの影響は OLSR のジッタの方が支配的であると考えられる．よって，以降ではメッセージの受信タイミングは OLSR のジッタの実現値によって変化するものとする．

2.6 リンク感知および隣接ノード探知

2.6.1 Hello メッセージの交換

OLSR のリンク感知および隣接ノード探知機能は Hello メッセージの交換によって実行される．Hello メッセージは以下のフィールドによって構成される（図 1）．

- 送信元アドレス，
 - リンクメッセージ．
- さらに，リンクメッセージフィールドには以下の情報が格納される．
- 隣接ノードアドレス，

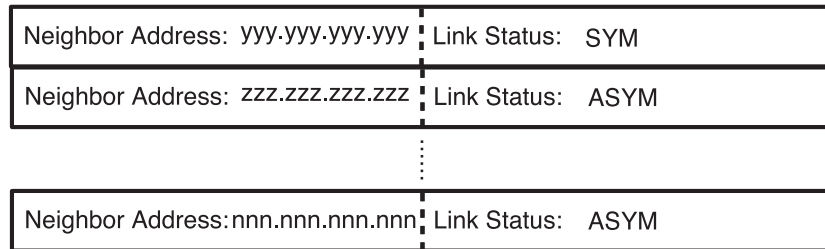


図 2 Neighbor Set の構造
Fig. 2 Structure of Neighbor Set.

- 送信元と隣接ノード間のリンク状態 .

これらの値は Neighbor Set に登録されている情報によって変化する .

また, Neighbor Set には以下の値が格納される (図 2).

- 隣接ノードアドレス ,
- リンク状態 .

Neighbor Set に隣接ノードが登録されていなければ送信する Hello メッセージ内のリンクメッセージには何も格納されず, 登録されていれば隣接ノードのアドレスと対応するリンクの状態がリンクメッセージフィールドに格納される . さらに, 複数の隣接ノードが登録されている場合は複数のリンクの情報がメッセージに格納される . ここで, リンク状態としては, 対称 (SYM), 非対称 (ASYM) が記録される .

また, Hello メッセージを受信したノードは, Neighbor Set を書き換えるが, それにあわせて, 2-hop Neighbor Set の情報も更新する . Hello メッセージのリンクメッセージフィールドに情報が格納されていた場合は, Hello メッセージの送信元がどのノードと隣接しているかを受信ノードが知ることができる . つまり, 受信ノードの視点で考えると, 送信元ノードを経由することで到達することのできる 2 ホップ隣接ノードがどのノードであるかを知ることができる . このように, 2 ホップ隣接ノードを判別することができたときに受信ノードの 2-hop Neighbor Set が更新される . この 2-hop Neighbor Set には以下の情報が格納される (図 3 参照).

- 2 ホップ隣接ノード
- 経由隣接ノード

なお, 2-hop Neighbor Set には, 対称リンクでつながっているノードのみが記載される .

このようにして, 各ノードは隣接ノードの情報を収集していく . これらの情報は経路表の

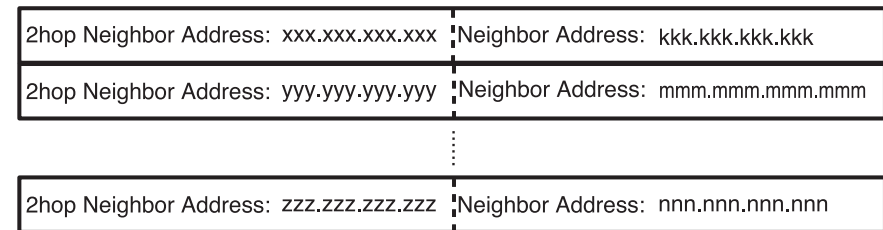


図 3 2-hop Neighbor Set の構造
Fig. 3 Structure of 2-hop Neighbor Set.

計算時に利用される .

ここで, Hello メッセージの交換により Neighbor Set が更新されていく様子を, 隣接ノード A, B がメッセージを交換する例を用いて例示的に説明する . なお, 一般に, このシーケンスは, Hello メッセージの受信タイミングによって変化する . しかし, OLSR の実装によっては, テーブル更新処理がジッタによるメッセージの受信タイミングに影響されないように, メッセージを受信直後に処理せず, 一定期間バッファリングしてから実際のテーブル更新を行う場合もある . そこで, これらの違いを説明するために

- 例 1 : Hello メッセージの受信直後にテーブルを更新する実装の場合,
- 例 2 : 一定期間バッファリングしてからテーブルを更新する実装の場合,

という 2 つの例を示す (図 4, 図 5 参照). ここでは, いずれの例でも, ノード A が Hello メッセージを送信する際のジッタがノード B のジッタよりも小さく, また, つねに両ノード間の無線通信が成功する場合を記述する .

2.6.1.1 例 1 : Hello メッセージを受信直後にテーブルを更新する実装の場合

- (1) 時刻 t_0 : 2 ノード A, B が OLSR ネットワークに組み込まれる . この時点では, A, B ともに Neighbor Set が空である .
- (2) 時刻 t_1 付近 : A, B ノードが互いに Hello メッセージを送信する . 両ノードとも Neighbor Set が空であるため, リンクメッセージフィールドには何も格納されない . まず, ジッタの影響でノード A がわずかに早く Hello メッセージを送信し, このメッセージがノード B に到達する . すると, ノード B は, 受信した Hello メッセージの送信元アドレスフィールドにノード A のアドレスが記載されているため, ノード A からノード B 方向の片方向の通信が可能であることを知る . ここで, ノード B の Neighbor Set には, 隣接ノードとして A, リンク状態として ASYM が記載された

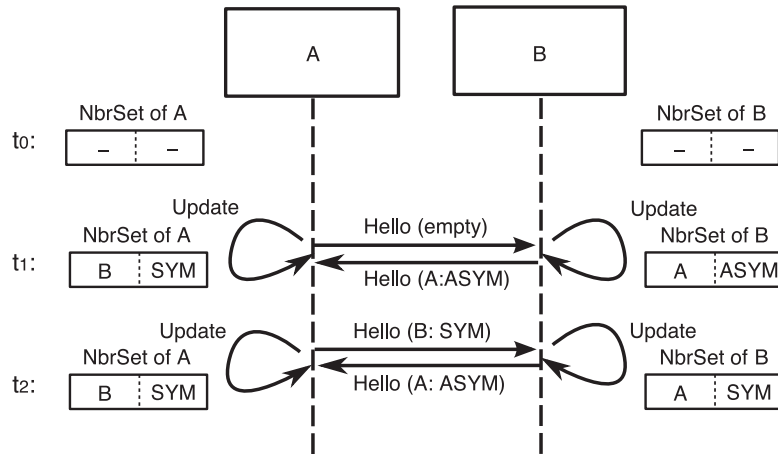


図 4 Hello メッセージ交換による Neighbor Set の更新 (例 1)
Fig. 4 Update Neighbor Set by Hello Message (case 1).

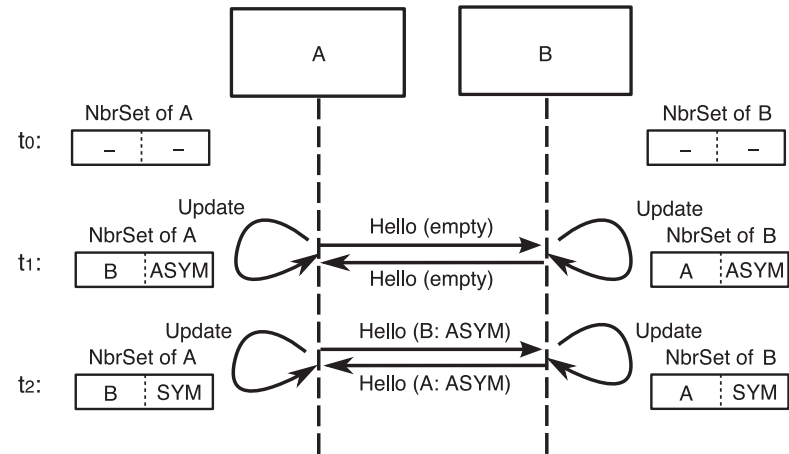


図 5 Hello メッセージ交換による Neighbor Set の更新 (例 2)
Fig. 5 Update Neighbor Set by Hello Message (case 2).

エントリが作成される．次に，ノード B が Hello メッセージを送信するが，この時点で B の Neighbor Set には，隣接ノード A が ASYM であるという情報が格納されているため，B が送信する Hello メッセージのリンクメッセージフィールドには，この情報が格納される．ノード A が B の Hello メッセージを受信すると，B から A 方向の通信が成功していることを知る．さらに，この Hello メッセージのリンクメッセージフィールドには，自身が B にとって ASYM であるという情報が格納されている．これにより，以前に A から送信した Hello メッセージが B に受信されていること，すなわち A から B 方向への通信が成功していることが分かる．よって，ノード A は AB 間のリンクが対称であることを認識し，Neighbor Set の B に関するエントリの状態が SYM に更新される．このようにして，B との対称隣接関係を確立することができる．

- (3) 時刻 t_2 付近：A の Neighbor Set には，隣接ノード B が SYM であるという情報が格納されているため，A が送信する Hello メッセージのリンクメッセージフィールドには，この情報が格納される．この Hello メッセージを B が受信すると時刻 t_1 での A の処理と同様に，Neighbor Set の B に関するエントリの状態が SYM に更新され，A との対称隣接関係を確立することができる．

2.6.1.2 例 2：一定期間バッファリングしてからテーブルを更新する実装の場合

- (1) 時刻 t_0 ：2 ノード A, B が OLSR ネットワークに組み込まれる．この時点では，A, B とともに Neighbor Set が空である．
- (2) 時刻 t_1 付近：A, B ノードが互いに Hello メッセージを送信する．例 1 の場合と異なり，ジッタの影響で B の Hello メッセージ送信前に A の Hello メッセージが到着していても，メッセージ送信前に Neighbor Set が更新されることがなく，メッセージ送信タイミングでは両ノードとも Neighbor Set が空である．そのため，リンクメッセージフィールドには何も格納されない．お互いに他方が送信した Hello メッセージを受信すると，Neighbor Set が更新される．ノード A は，受信した Hello メッセージの送信元アドレスフィールドにノード B のアドレスが記載されているため，ノード B からノード A 方向の片方向の通信が可能であることを知る．そこで，ノード A の Neighbor Set には，隣接ノードとして B，リンク状態として ASYM が記載されたエントリが作成される．ノード B の Neighbor Set も同様に更新される．
- (3) 時刻 t_2 付近：A の Neighbor Set には，隣接ノード B が ASYM であるという情報が格納されているため，A が送信する Hello メッセージのリンクメッセージフィールドには，この情報が格納される．B が送信する Hello メッセージにも，同様に隣接ノード

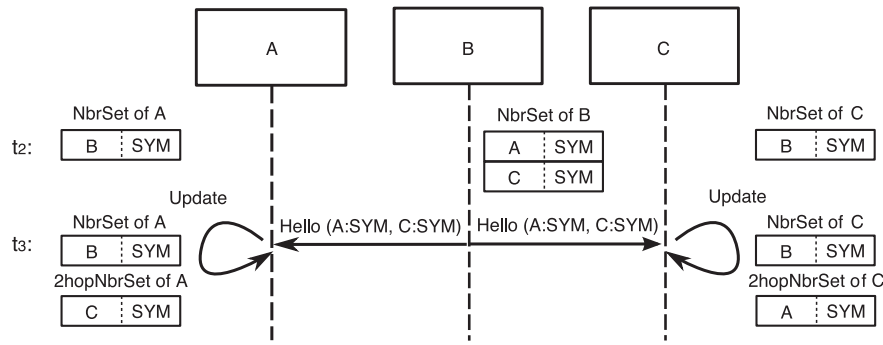


図 6 Hello メッセージ交換による 2-hop Neighbor Set の更新
Fig. 6 Update 2-hop Neighbor Set by Hello Message.

ド A が ASYM であるという情報が格納される。ノード A が B の Hello メッセージを受信すると、前回の受信時と同じように、B から A 方向の通信が成功していることを知る。さらに、この Hello メッセージのリンクメッセージフィールドには、自身が B にとって ASYM であるという情報が格納されている。これにより、以前に A から送信した Hello メッセージが B に受信されていること、すなわち A から B 方向への通信が成功していることが分かる。このようにして、ノード A は AB 間のリンクが対称であることを認識し、Neighbor Set の B に関するエントリの状態が SYM に更新される。ノード B についても同じような処理が行われ、A との対称隣接関係を確立することができる。

さらに、2-hop Neighbor Set に情報が登録される様子を、A, B, C の Hello メッセージ交換を例にとり以下に示す (図 6 参照)。ここでは、A と B, B と C が隣接ノードであるとし、A, C 間では直接の通信ができないものとしている。

- (1) 時刻 t_2 付近：図 4, 図 5 の例と同様に、各ノードが Hello メッセージを 2 回交換し、ノード A, B 間およびノード B, C 間で対称隣接関係が確立される。
- (2) 時刻 t_3 付近：ノード B が送信する Hello メッセージのリンクメッセージフィールドには、A が SYM, C も SYM であるという情報が格納される。ノード A がこの Hello メッセージを受信すると、B の隣接ノードが C であるということを知る。そこで、ノード A の 2-hop Neighbor Set には、2 ホップ隣接ノードとして C, 経由隣接ノードとして B が記載されたエントリが作成される。同様に、ノード C の 2-hop

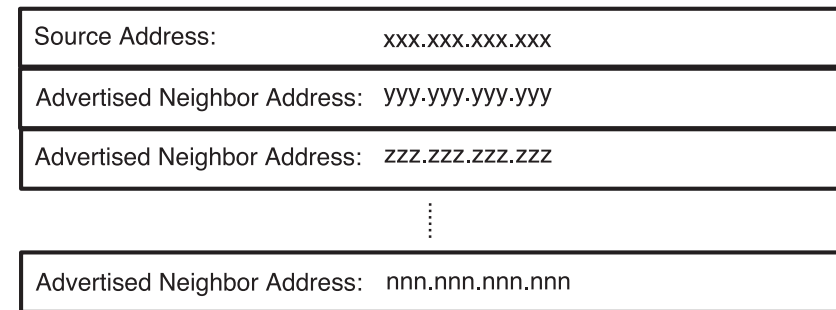


図 7 TC メッセージの構造
Fig. 7 Structure of TC Message.

Neighbor Set には、2 ホップ隣接ノードとして A が記録される。

2.7 トポロジ開示

トポロジ開示の機能は TC メッセージによって実行される。

TC メッセージは以下のフィールドによって構成される (図 7 参照)。

- 送信元アドレス,
 - 広告隣接ノードアドレス.
- 広告隣接ノードアドレスフィールドには、メッセージ送信元ノードの隣接ノードのうちいくつかのノードのアドレスが格納される。アドレスを格納するノードの選択条件は OLSR の設定によって異なるが、本論では対称隣接ノードをすべて選択するという条件のもと格納するものとする。

TC メッセージを受け取ったノードは、自身の Topology Set を書き換える。

Topology Set では、主に以下の情報が保持される (図 8 参照)。

- 宛先アドレス,
 - 宛先直前アドレス.
- TC メッセージを受け取ったノードは、メッセージの広告隣接ノードアドレスを Topology Set の宛先アドレスに、送信元アドレスを宛先直前アドレスに、それぞれ記載する。この情報は経路表の計算時に使用される。また、この TC メッセージは、隣接ノードだけでなく全ノードに転送される。そのため、隣接ノード情報を遠隔ノード間で交換することができる。
- TC メッセージの交換により、Topology Set が書き換わっていく様子を以下に例示的に示す (図 9 参照)。

Destination Address: xxx.xxx.xxx.xxx	Last Address: kkk.kkk.kkk.kkk
Destination Address: yyy.yyy.yyy.yyy	Last Address: mmm.mmm.mmm.mmm
⋮	
Destination Address: zzz.zzz.zzz.zzz	Last Address: nnn.nnn.nnn.nnn

図 8 Topology Set の構造
Fig.8 Structure of Topology Set.

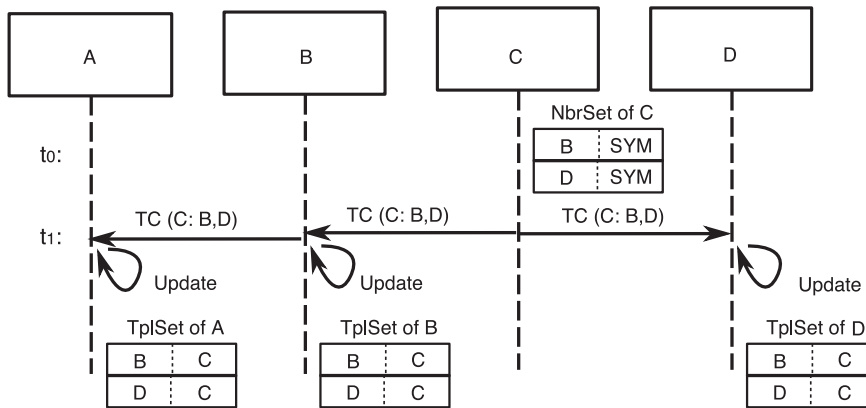


図 9 TC メッセージ交換のシーケンス
Fig.9 Update Topology Set.

- (1) 時刻 t_0 付近: 各ノードの Hello メッセージの交換により, ノード A, B 間および B, C 間, C, D 間で対称隣接関係が確立される.
- (2) 時刻 t_1 付近: ノード C が送信する TC メッセージの広告隣接ノードアドレスには, 自身の隣接ノードである B と D が格納される. まず, ノード B と D がこのメッセージを受信し, 宛先アドレスに B と D, 宛先直前アドレスに C が記載されたエントリを Topology Set に登録する. さらに, ノード B はただちにこの C から送信された TC メッセージをノード A に転送する. そこでノード A も同様のエントリを Topology

Destination Address: xxx.xxx.xxx.xxx	Next Hop Address: kkk.kkk.kkk.kkk	Distance: a
Destination Address: yyy.yyy.yyy.yyy	Next Hop Address: mmm.mmm.mmm.mmm	Distance: b
⋮		
Destination Address: zzz.zzz.zzz.zzz	Next Hop Address: nnn.nnn.nnn.nnn	Distance: c

図 10 経路表の構造
Fig.10 Structure of routing table.

Set に登録する. また, 他のノードが送信した TC メッセージによっても, 同様の処理が行われる.

なお, TC メッセージは, MPR ノードのみが送信する. 上記の例では, B および C が A もしくは D から送信されたデータを転送することで全ノード間のデータ通信が可能となるため, B と C が MPR ノードとして選択される.

2.8 情報保持期限

MANET の環境では, 無線の状態によってリンクの通不通が変化する. そのため, OLSR には一定期間ごとにネットワークの情報をリフレッシュする機能が備わっている. Hello, TC の両メッセージによって交換される情報にはすべて時刻が記録されており, Neighbor Hold Time という有効期間が過ぎるとリセットされる. その間に次の Hello メッセージが届かなかった場合は Neighbor Set および 2-hop Neighbor Set に保持されている情報が破棄され, 届いた場合には有効期限が更新される. 同様に, TC メッセージによって交換される情報は Topology Hold Time という情報保持期限を持つ.

2.9 経路表作成

経路表は以下の項目からなる (図 10 参照).

- 宛先アドレス,
- 次ホップアドレス,
- 距離 (ホップ数).

この経路表は, Neighbor Set, 2-hop Neighbor Set, Topology Set のいずれかの更新時に, 以下の手順で計算される.

- (1) 前回計算した経路表をリセットする.
- (2) 隣接ノードのうち, 対称隣接関係にあるノードを経路表に追加する. 隣接ノードには直接到達することができるため, 宛先アドレス, 次ホップアドレスともに隣接ノード

のアドレスを登録する．距離は 1 となる．

- (3) 2 ホップ隣接ノードを追加する．2-hop Neighbor Set に格納されている情報より，2 ホップ隣接ノードのアドレスを宛先アドレスに，1 ホップ隣接ノードのアドレスを次ホップアドレスとして登録する．距離は 2 となる．
- (4) 遠隔ノードを追加する．経路表の宛先アドレスに記載されているアドレスが，Topology Set の宛先直前アドレスとして登録されていれば，そのノードを経由して，Topology Set 上の対応する宛先アドレスに到達することができる．たとえば，ノード A のステップ 3 以前の経路表作成過程で，宛先 C に対して次ホップ B が登録されたとする．さらに，Topology Set には，宛先 D に対応する宛先直前アドレス C が登録されていたとする．このとき，ノード A から次ホップ B に送信されたパケットは，ノード C を経由して宛先 D に到達できることが分かる．そこで，宛先 D，次ホップ B，距離 3 のエントリが経路表に追加される．このようにして順次 Topology Set のエントリを経路表に追加していく．

3. OLSR の経路表作成モデル

3.1 概要

本章では，OLSR の経路表完成時間を評価するための数理モデルを構成するうえで考慮すべき項目，ならびに性能評価シナリオに共通するパラメータについて記述する．

前章に述べたように，OLSR の経路表は各ノード間で送受信される Hello メッセージおよび TC メッセージに記述されている情報に基づいて作成されていく．このとき，経路情報を最新のものに保つため，情報には Neighbor Hold Time および Topology Hold Time といわれる保持期限が設けられており，一定時間以上経過した情報は破棄される．経路表作成時間には，これらの要素が相互に関連しながら影響することになる．これらを考慮して，モデルを構成する際に考慮すべきパラメータおよび仮定について以下に記す．

3.2 パラメータ

- (1) ノード数：ネットワーク内のノードの集合を \mathcal{N} で表すことにする．また，ノード数を $N = |\mathcal{N}|$ とする．以降では，例として $\mathcal{N} = \{A, B, C, D\}$ ， $N = 4$ の場合を記述する．
- (2) 各ノード間の無線通信の成功確率：各ノード間は無線通信を行うが，その通信が成功するか失敗するかは確率的に定まるものと考え，ノード i, j ($i, j \in \mathcal{N}$) 間の無線通信成功確率を r_{ij} とし，失敗確率を $r_{ij}^* = 1 - r_{ij}$ と表すことにする．また，これ

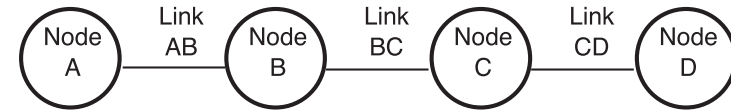


図 11 ネットワークトポロジ
Fig. 11 Network topology.

らの確率は対称であると考え， $r_{ij} = r_{ji}$ とする．

- (3) Hello メッセージの送信時間間隔：メッセージ送信間隔は，先に述べたように，OLSR プロトコルの実装に依存して定まるが，ここでは以下のように設定する．
 - Hello Interval: $T_{\text{Hello}} = 2$ [秒]
 - (4) TC メッセージの送信間隔：同様に以下のように設定する．
 - TC Interval: $T_{\text{TC}} = 4$ [秒]
 - (5) 接続情報の保持期限：情報保持期限も，OLSR プロトコルの実装に依存して定まるが，RFC ではメッセージ送信間隔の 3 倍の値に設定することが推奨されている．そこで，ここでは以下のように設定する．
 - Neighbor Hold Time: $T_{\text{Neighbor}} = 6$ [秒].
 - (6) トポロジ情報の保持期限：同様に以下のように設定する．
 - Topology Hold Time: $T_{\text{Topology}} = 12$ [秒].
- #### 3.3 仮定
- (7) ネットワークのトポロジ：本論では，数理的な解析を可能とするため，一次元のネットワークトポロジのみを考えることにする．先に述べたように，ここでは 4 ノードの例を考えているため，図 11 に示されるような，A, B, C, D の 4 ノードが一次的に配置されたトポロジとなる．ここで，隣り合った 2 ノードのみが無線を通じて直接ができるものとする．また，これらの 4 ノードは時刻 0 で同時にネットワークに組み込まれる（同時に電源が投入される）ものとする．このとき，ノード A に着目し，時刻 0 からノード A の経路表が完成するまでの時間の期待値に関する評価を行う．さらに，直線型の一次元のネットワークトポロジでは，最両端（4 ノードの場合は A, D）を除く全ノードがメッセージを中継しないと，全ノード間で互いにメッセージの交換を行うことができない．そのため，これらのノード（4 ノードの場合は C, B）は MPR となり，TC メッセージを送信する．
 - (8) TC メッセージの到達確率：本来であれば，あるノードが送信する TC メッセージが

遠隔ノードに到達する確率は、経由した全ノード間のリンクの状態に依存する。しかし、解析を可能とするためノード C が送信する TC メッセージをノード A が受信できる確率を確率 $\gamma = r_{AB}r_{BC}r_{CD}^2$ で表現できるものと仮定する。この確率は、ノード C-D 間で Hello メッセージを 2 回交換することで対称隣接関係を構築し、その後、ノード C からノード B へ、ノード B からノード A へと TC メッセージの転送が成功する確率を表している。

- (9) Neighbor Hold Time の計測方法と満了時の情報更新方法：これも実装に依存するが、本論では簡略化のため、各ノードは一定時間間隔 $T_{Neighbor}$ ごとに情報を更新するものとした。また、更新対象とする情報は Hello メッセージの交換により更新される各ノードの Neighbor Set および 2-hop Neighbor Set とし、これらのテーブルに記載されているリンク状態が対称リンク (SYM) であるものも非対称リンク (ASYM) であるものもすべてそのリンク情報を無効状態 (LOST) にリセットするものとする。
- (10) Topology Hold Time の計測方法と満了時の情報更新方法：これも実装に依存するが、本論では簡略化のため、各ノードは一定時間間隔 $T_{Topology}$ ごとに情報を更新するものとした。ここで更新対象となるのは、TC メッセージの交換により更新される各ノードの Topology Set に記載されているエントリである。

なお、OLSR の多くの実装では、Neighbor Set および 2-hop Neighbor Set, Topology Set の各エントリごとに情報保持期限のタイマがセットされ、情報が更新される。提案モデルでは、Neighbor Hold Time, Topology Hold Time のタイマが一定間隔で満了すると仮定しているため、実際の OLSR より情報をリセットするイベントが多く発生する。よって、提案モデルは経路表作成時間を長く見積もることになる。これは、ネットワーク設計の観点から考えると、安全側の見積りとなる。

以上をまとめて、時刻 0 から始まり、時刻 12 秒までの各ノードにおけるメッセージ送受信のタイミングおよび Neighbor Hold Time, Topology Hold Time の更新のタイミングを図示すると図 12 のようになる。12 秒以降はこの関係が繰り返されていくことになる。

3.4 非斉時マルコフ連鎖

図 12 に示されるように、各時刻でイベントが発生するが、発生するイベントの種類により、ネットワークの状態変化は異なっている。 $0 \leq t \leq 12$ について、発生時刻 t を発生するイベントの種類によって分類してみると以下のようになる。

- $t \in \{1, 3, 5, 7, 9, 11\}$: 発生イベントなし,
- $t \in \{2, 10\}$: Hello メッセージ受信発生,

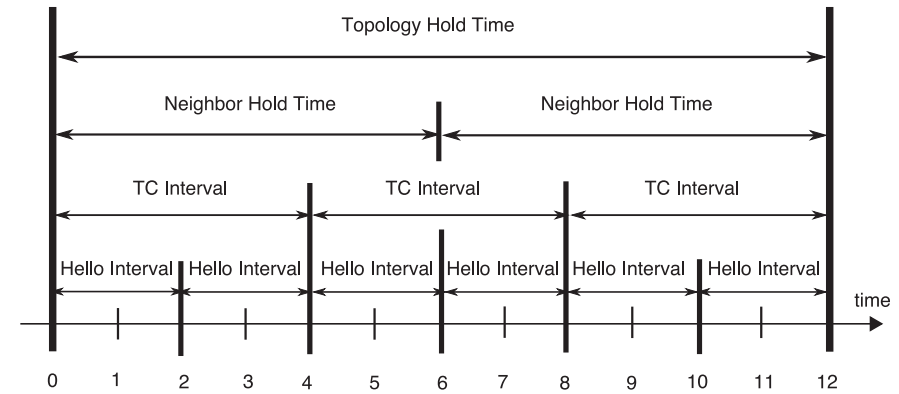


図 12 OLSR のタイミングチャート

Fig. 12 Timing chart of OLSR.

- $t \in \{4, 8\}$: Hello メッセージ受信および TC メッセージの受信がともに発生,
- $t \in \{6\}$: Hello メッセージ受信および Neighbor Hold Time 経過イベントがともに発生,
- $t \in \{12\}$: Hello メッセージ受信, TC メッセージ受信および Neighbor Hold Time 経過イベント, Topology Hold Time 経過イベントがともに発生.

$t > 12$ についてはこの繰返しになるので詳細は省略をする。

ノード A の経路表を作成するには、ノード A の Neighbor Set, 2-hop Neighbor Set, および Topology Set に示される情報に加えて、各時刻でどのようなイベントが発生したのかという情報が分かればよい。また、どのようなイベントが起きるかは各時刻で確率的に定まるので、このネットワーク全体の動作はマルコフ連鎖として記述できることが分かる。ただし、時刻により発生するイベントの種類が異なるため、時間的に一様なマルコフ連鎖にはならない。そのため、時刻ごとに推移確率が異なる非斉時マルコフ連鎖としてモデル化を行う。このとき、ノード A の経路表が完成をした状態をこのマルコフ連鎖の吸収状態とし、その他の状態は一時的状態としてあつかう。したがって、ノード A の経路表完成時間の期待値は、この吸収状態を持つ非斉時マルコフ連鎖において、ある初期状態から出発して吸収状態にいたるまでの平均吸収時間として得られることになる。

3.5 ネットワークの状態定義

ネットワークの状態を 4 変数 (Y_1, Y_2, Y_3, Y_4) で表すことにする。ここで、

- $Y_1 \in \{LOST, ASYM, SYM\}$: ノード A の Neighbor Set の状態

- $Y_2 \in \{\text{LOST}, \text{ASYM}, \text{SYM}\}$: ノード B の Neighbor Set の状態
 - $Y_3 \in \{\text{KNOWN}, \text{UNKNOWN}\}$: ノード A の 2-hop Neighbor Set
 - $Y_4 \in \{\text{NONE}, \text{RECEIVE}\}$: ノード A の Topology Set
- である。各変数の持つ値の意味は以下のとおりである。
- Y_1 : (ノード A の Neighbor Set の状態)
 - LOST : ノード A がいずれのノードにもつながっていない状態。
 - ASYM : ノード A がノード B からの Hello メッセージを受信し、ノード B から自身への片方向の無線リンクが有効であることを認識した状態。
 - SYM : ノード A, B 間で互いに Hello メッセージを交換し、リンク AB が双方向に通信可能であることをノード A が認識した状態。
 - Y_2 : (ノード B の Neighbor Set の状態)
 - LOST : ノード B がいずれのノードにもつながっていない状態。
 - ASYM : ノード B がノード C からの Hello メッセージを受信し、ノード C から自身への片方向の無線リンクが有効であることを認識した状態。
 - SYM : ノード B, C 間で互いに Hello メッセージを交換し、リンク BC が双方向に通信可能であることをノード B が認識した状態。
 - Y_3 : (ノード A の 2-hop Neighbor Set の状態)
 - UNKNOWN : リンク BC が双方向に通信可能であることを、ノード A が認識していない状態。
 - KNOWN : リンク BC が双方向に通信可能であることを、ノード A が認識した状態。つまり、ノード B が送信した Hello メッセージ内に C が対称隣接ノードであるという情報が格納され、かつそのメッセージをノード A が受信できた状態。この状態に推移するということは、ノード A の 2-hop Neighbor Set にノード C が記載されるということである。
 - Y_4 : (ノード A の Topology Set の状態)
 - NONE : ノード A がノード D の TC メッセージを受信していない状態。
 - RECEIVE : ノード A がノード D の TC メッセージを受信し、ノード D の経路を確立した状態。

この条件のもとで、ネットワークの状態 (Y_1, Y_2, Y_3, Y_4) がとりうる値の可能な値な組合せをすべて調べると以下に示す 12 通りの状態に集約されることが分かる。

$$s_1 = (\text{LOST}, \text{LOST}, \text{UNKNOWN}, \text{NONE}),$$

$$s_2 = (\text{LOST}, \text{ASYM}, \text{UNKNOWN}, \text{NONE}),$$

$$s_3 = (\text{LOST}, \text{SYM}, \text{UNKNOWN}, \text{NONE}),$$

$$s_4 = (\text{ASYM}, \text{LOST}, \text{UNKNOWN}, \text{NONE}),$$

$$s_5 = (\text{ASYM}, \text{ASYM}, \text{UNKNOWN}, \text{NONE}),$$

$$s_6 = (\text{ASYM}, \text{SYM}, \text{UNKNOWN}, \text{NONE}),$$

$$s_7 = (\text{SYM}, \text{LOST}, \text{UNKNOWN}, \text{NONE}),$$

$$s_8 = (\text{SYM}, \text{ASYM}, \text{UNKNOWN}, \text{NONE}),$$

$$s_9 = (\text{SYM}, \text{SYM}, \text{UNKNOWN}, \text{NONE}),$$

$$s_{10} = (\text{SYM}, \text{SYM}, \text{KNOWN}, \text{NONE}),$$

$$s_{11} = (\text{SYM}, \text{SYM}, \text{UNKNOWN}, \text{RECEIVE}),$$

$$s_{12} = (\text{SYM}, \text{SYM}, \text{KNOWN}, \text{RECEIVE}).$$

図 13 にこれらの状態と各ノード間のリンクの接続状態、およびノード A の Neighbor Set, ノード B の Neighbor Set, ノード A の 2-hop Neighbor Set, ノード A の Topology Set の状態の対応を示す。状態 s_{12} がノード A の経路表が完成した状態に対応している。

以上より、ネットワークの状態空間をあらためて $S = \{s_i\}_{i=1}^{12}$ と定義し、状態空間 S 上の値をとる確率変数列 $\{X_n\}_{n=0}^{\infty}$ をマルコフ連鎖として表現することにする。

3.6 状態推移確率行列

考えているマルコフ連鎖 $\{X_t\}_{t=0}^{\infty}$ の状態推移確率を $\{P_t\}_{t=0}^{\infty}$ とする。

ネットワークの状態変化を引き起こすイベントの発生時刻は、 $0 \leq t \leq 12$ については 3.4 節で述べたように分類できた。この分類に従って 5 種類の時刻に対応するマルコフ連鎖の推移確率行列を考えることにする。また、ネットワークの状態推移の時間的構造は 12 秒を単位として繰返し構造になっていることを考慮して、あらためてイベント発生時刻の集合を次のように分類し定義する。

- $\mathcal{I} = \{t + 12m | t \in \{1, 3, 5, 7, 9, 11\}, m = 0, 1, 2, \dots\}$: 発生イベントなし,
- $\mathcal{A} = \{t + 12m | t \in \{2, 10\}, m = 0, 1, 2, \dots\}$: Hello のメッセージ受信発生,
- $\mathcal{B} = \{t + 12m | t \in \{4, 8\}, m = 0, 1, 2, \dots\}$: Hello メッセージおよび TC メッセージの受信がともに発生,
- $\mathcal{C} = \{t + 12m | t \in \{6\}, m = 0, 1, 2, \dots\}$: Hello メッセージ受信および Neighbor Hold Time 経過イベントがともに発生,
- $\mathcal{D} = \{t + 12m | t \in \{12\}, m = 0, 1, 2, \dots\}$: Hello メッセージ受信, TC メッセージ受信および Neighbor Hold Time, Topology Hold Time 経過イベントがともに発生。

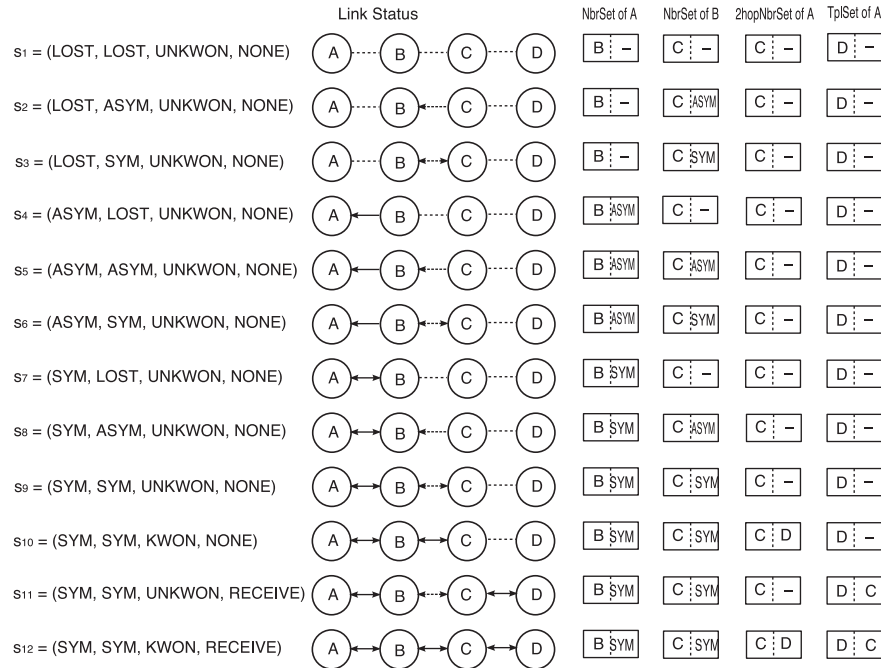


図 13 ネットワークの状態
Fig. 13 State of network.

このとき、 $t \in I$ については明らかに $P_t = I$ (単位行列) となる。

また、前章で述べたとおり、OLSR のジッタの影響で Hello メッセージの到着順によってリンク感知の際の状態推移が異なる。そこで、この違いを表現するために以下の確率を導入する。

$$\delta_{ij} = P(X_i \leq X_j), \quad \delta_{ji} = P(X_i > X_j), \quad i, j \in \mathcal{N}.$$

ここで、 X_i, X_j は、それぞれノード i, j のジッタ値を表す確率変数とする。つまり、 δ_{ij} はノード i の Hello メッセージが、ノード j のメッセージよりも早く送信される確率を表す。なお、 $\delta_{ij} + \delta_{ji} = 1$ となる。これにより、前節で述べた経路表計算タイミングによる実装の違いを表現することができる。先に例として用いたメッセージ受信直後に経路表計算を行う実装は、これらを等確率であるとし、

$$\delta_{AB} = \delta_{BA} = \frac{1}{2}, \quad \delta_{BC} = \delta_{CB} = \frac{1}{2}$$

とすることで表現できる。それに対して、受信タイミングに依存しないようにメッセージをバッファリングする実装は、

$$\delta_{AB} = 0, \quad \delta_{BA} = 1, \quad \delta_{BC} = 0, \quad \delta_{CB} = 1$$

とすることで表現できる。

前節に述べた OLSR の動作と、本節に述べたいいくつかの仮定に基づいて推移確率行列を構成すると付録 A.1 のようになる。

これらの推移確率行列より、マルコフ連鎖 $\{X_t\}_{t=0}^{\infty}$ は吸収状態 s_{12} と一時的状態 $\{s_i\}_{i=1}^{11}$ を持ち、 $t \rightarrow \infty$ においてはいかなる初期状態から出発しても確率 1 で吸収状態に到達することが分かる。

3.7 平均吸収時間

時刻 T におけるネットワークの状態確率 π_T は、初期状態ベクトルを π_0 とし、前節に示した推移確率行列 P_t を用いて

$$\pi_T = \pi_0 \prod_{t=1}^T P_t \tag{1}$$

と表すことができる。この関係を用いて、時刻 t においてノード A の経路表が完成する確率を計算することができるが、考えているマルコフ連鎖は時刻とともに推移確率行列 P_t の形が変化する非斉時マルコフ連鎖なので、その計算は煩雑になる。そこで、本論では平均吸収時間のみに着目し、以下のような手順で計算をすることにする。本マルコフ連鎖の推移確率構造が 12 秒を単位として変化することに注目し、時間単位が 12 秒であり、次のような推移確率行列を持つ時間的に一様なマルコフ連鎖 $\{X_n^*\}_{n=0}^{\infty}$ を考え、その平均吸収時間を求めることにする。

$$P = \prod_{i=1}^{12} P_i. \tag{2}$$

この時間的に一様なマルコフ連鎖 $\{X_n^*\}_{n=0}^{\infty}$ は、マルコフ連鎖 $\{X_t\}_{t=0}^{\infty}$ と同じ状態空間 S を持ち、同じ吸収状態 s_{12} と同じ一時的状態 $\{s_i\}_{i=1}^{11}$ を持つことは明らかである。マルコフ連鎖 $\{X_n^*\}_{n=0}^{\infty}$ において、与えられた初期状態から出発して、吸収状態 s_{12} に吸収されるまでの平均吸収時間を求めるため、推移確率行列 P を以下のように一時的状態と吸収状態

に分割する .

$$P = \begin{matrix} & s^* & s_{12} \\ s^* & \begin{pmatrix} Q & R \\ O & 1 \end{pmatrix} \\ s_{12} & \end{matrix} \quad (3)$$

ただし, $s^* = \{s_i\}_{i=1}^{11}$ は一時的状態の集合とする . このときこのマルコフ連鎖の基本行列 M は

$$M = (I - Q)^{-1} = \sum_{m=0}^{\infty} Q^m \quad (4)$$

で与えられ, 平均吸収時間ベクトル $\tau = \{\tau(s_i)\}_{i=1}^{12}$ は

$$\tau = M\mathbf{1} \quad (5)$$

となることはよく知られている⁶⁾. ここで, $\tau(s_i)$ は初期状態 s_i ($i = 1, 2, \dots, 12$) から出発して吸収状態 s_{12} に到達するまでの平均時間, すなわち平均吸収時間を表しており, $\mathbf{1} = (1, 1, \dots, 1)^t$ である .

以上より, 初期状態 s_1 から経路表が完成する状態 s_{12} に到達するまでの平均時間は式 (5) より計算される $\tau(s_1)$ を 12 倍したもので, すなわち $T_{create} = 12 \times \tau(s_1)$ 秒となる .

4. 評価例

4.1 送受信成功率を変更させた場合の評価

前章で述べたモデルを使用した経路表作成時間の評価例として, 送受信成功率を変更する 4 つの評価シナリオを考える .

- (1) 評価シナリオ 1-1: 以下 3 つのシナリオでは特定のリンクの強度を変化させる . ここでは r_{AB} の強度を 0.1 から 1.0 へと変化させる . このシナリオは, ある特定のノードのみの無線環境が悪くなる場合を想定している . たとえば, あるノードのみが全体から離れていき, 到達電波が弱くなる場合などが考えられる .
- (2) 評価シナリオ 1-2: 評価シナリオ 1-1 と同様に r_{BC} の強度を変化させる .
- (3) 評価シナリオ 1-3: 評価シナリオ 1-1, 1-2 と同様に r_{CD} の強度を変化させる .
- (4) 評価シナリオ 2: このシナリオでは, 全リンクの強度を変化させる, すなわち, $r_{ij} = 0.1, 0.2, \dots, 1.0$ ($i, j \in \mathcal{N}$) と均等に变化させる . このシナリオは, たとえば雨が降って全体の無線状況が悪化する場合のように, ネットワーク全体の環境が

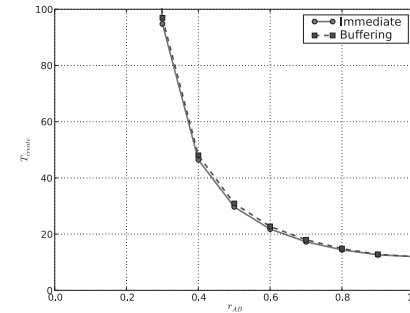


図 14 シナリオ 1-1 の経路表作成時間の推移 (x: r_{AB} , y: T_{create})
Fig. 14 Routing table calculation time in scenario 1-1.

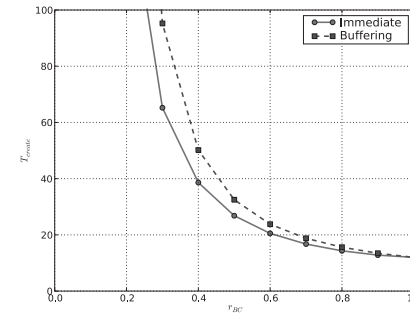


図 15 シナリオ 1-2 の経路表作成時間の推移 (x: r_{BC} , y: T_{create})
Fig. 15 Routing table calculation time in scenario 1-2.

均一に変化するような状況を想定している .

まず, 評価シナリオ 1-1 ~ 1-3 の経路表作成時間評価を, それぞれ図 14, 図 15, 図 16 に示す . ここで, 図中の「Immediate」はメッセージ受信直後に経路表を計算する場合, 「Buffering」はメッセージ受信タイミングに依存しないようにバッファリングしてから経路表計算する場合の評価値を示している . 評価シナリオ 1-1 ~ 1-3 の結果を比較することにより, どの無線リンクの強度が最も経路表の作成時間に影響を与えるかを評価することができる . 今回の評価では, ノード A から一番近いリンク AB の強度が最も経路表作成時間に影響を与えることが分かる . また, Hello メッセージの受信タイミングによる影響はあまり見られませんが, リンク BC の強度を変化させたときには, この影響が大きくなることが分かる .

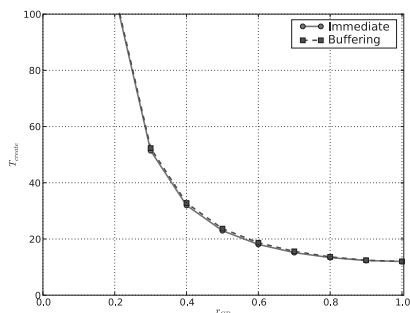
図 16 シナリオ 1-3 の経路表作成時間の推移 ($x: r_{CD}, y: T_{create}$)

Fig. 16 Routing table calculation time in scenario 1-3.

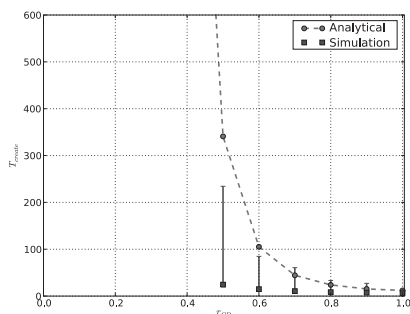
図 17 シナリオ 2 の経路表作成時間の推移 ($x: r_{ij}, y: T_{create}$)

Fig. 17 Routing table calculation time in scenario 2.

さらに、図 17 に評価シナリオ 2 の解析結果を示す。ここでは、提案モデルの評価のためシミュレーション実験による結果も合わせて示した。図中の「Analytical」が提案手法、「Simulation」がシミュレーション実験によって得られた結果である。シミュレーション実験から得られたデータとして、各点で 1,000 回試行後の平均値、最小値、最大値をグラフにプロットしている。なお、このシミュレーションでは、IP 層では OLSR を、MAC 層でも IEEE802.11 のプロトコルの動作を模擬したモデルを使用した。ここで、提案手法、シミュレーション実験ともメッセージ受信直後に経路表を計算する実装を用いた。提案手法とシミュレーション実験の結果を比較すると、送受信成功確率が小さくなると過度に大きく評価値を見積もる傾向にあるが、提案手法から得られた評価値が安全側の見積りであることが

確認できる。また、今回のシナリオの場合は、提案手法の評価値がシミュレーション実験での最大値に近い値となっている。

4.2 上記以外のパラメータを変更させた評価を行う場合の留意点

- (1) ノード数：ノード数を変更する場合は、前節で述べた例よりも TC メッセージを送信する MPR ノードが増えるため、TC の受信状態に関する状態数が増えることになる。ただし、TC メッセージ関連する状態のみを追加すればよいため、状態数が組合せ的に増大することなく、モデルを構成することができる。
- (2) メッセージ送信間隔と情報保持期限：これらのパラメータを変更すると状態推移のタイミングが変わるため、推移確率行列を再構成する必要がある。その際、これらのパラメータの最小公倍数

$$T_{LCM} = \text{lcm}(T_{\text{Hello}}, T_{\text{TC}}, T_{\text{Neighbor}}, T_{\text{Topology}})$$

が非斉時マルコフ連鎖の繰返し時間となる。よって、吸収時間を計算するために使用するマルコフ連鎖は、1 ステップ T_{LCM} 秒の時間的に一様なマルコフ連鎖となる。なお、これらのパラメータは、状態数に影響を与えることなく変更することができる。

5. 結 論

本論では、代表的な MANET 用のプロアクティブ型ルーティングプロトコルである OLSR の性能を評価するための数理解析モデルを提案した。性能評価指標として経路表作成時間を用い、時間的に非斉時なマルコフ連鎖を用いてモデル化を行った。モデルの解析過程で、非斉時なマルコフ連鎖から時間的に一様なマルコフ連鎖を構成するという工夫を施した。さらに、評価例として、無線リンクの強度（無線通信成功確率）が経路表の構築過程に与える影響を数値例を用いて示すことができた。

本論で対象とした経路表作成時間は、シミュレーションによる評価法では各試行から得られた最小値と最大値の幅が大きくなり、統計的に妥当な平均値を得るためには多くの計算時間が必要となる。それに対して、提案手法は確率過程を数学的に解析することで評価指標を得られるという利点がある。しかし、無線通信成功確率が低い状況では過度に作成時間を多めに見積もってしまうという欠点もある。

今後の課題としては、1 度経路表が完成した後に経路を維持できる時間など、新たな性能評価指標を計算することがあげられる。これに対しては、制約付きマルコフ連鎖の理論を適用するなど、提案手法をさらに解析するという解決策が考えられる。また、今回想定した直線型のネットワークポロジを 2 次元的な広がりを持つメッシュ型などにモデルを拡張する

ことも課題としてあげられる。2次元的な拡張を行うためには、2次元のメッセージ交換による状態変化だけでなく、複数の MPR 候補から MPR を選択するアルゴリズムを考慮する必要がある。

参考文献

- 1) Clausen, T., Hansen, G., Christensen, L. and Behrmann, G.: The Optimized Link State Routing Protocol, Evaluation through Experiments and Simulation, *IEEE Symposium on "Wireless Personal Mobile Communications"* (Sep. 2001).
- 2) Clausen, T. and Jacquet, P.: Optimized Link State Routing Protocol (OLSR), *Request for Comments 3626* (2003).
- 3) Corson, S. and Macker, J.: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, *Request for Comments, Vol.2501* (1999).
- 4) IEEE Computer Society: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Technical report, Institute of Electrical and Electronics Engineers (2007).
- 5) Laouti, A., Muhlethaler, P., Najid, A. and Plakoo, E.: Simulation Results of the OLSR Routing Protocol for Wireless Network, *Med-Hoc-Net 2002* (2002).
- 6) 森村英典, 高橋幸雄: マルコフ解析, 日科技連 (1979).

付 録

A.1 推移確率行列

$t \in \mathcal{A}$ について

$$P_t = \begin{pmatrix} r_{AB}^* & \delta_{CB} r_{AB}^* & \delta_{BC} r_{AB}^* & \delta_{BA} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{AB} r_{AB}^* & \delta_{AB} \delta_{BC} r_{AB}^* & 0 & 0 & 0 \\ 0 & r_{AB}^* & r_{AB}^* & 0 & \delta_{BA} r_{AB}^* & \delta_{BA} r_{AB}^* & 0 & \delta_{AB} r_{AB}^* & \delta_{AB} r_{AB}^* & 0 & 0 & 0 \\ 0 & 0 & r_{AB}^* & 0 & 0 & r_{AB} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{AB}^* & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{AB}^* & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{BC}^* & \delta_{CB} r_{BC}^* & \delta_{BC} r_{BC}^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{BC}^* & r_{BC} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{AB}^* & r_{AB} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{AB}^* & r_{AB} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$t \in \mathcal{B}$ について

$$P_t = \begin{pmatrix} r_{AB}^* & \delta_{CB} r_{AB}^* & \delta_{BC} r_{AB}^* & \delta_{BA} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{AB} r_{AB}^* & \delta_{AB} \delta_{BC} r_{AB}^* & \delta_{AB} \delta_{BC} r_{AB}^* & 0 & 0 & 0 \\ 0 & r_{AB}^* & r_{AB}^* & 0 & \delta_{BA} r_{AB}^* & \delta_{BA} r_{AB}^* & 0 & \delta_{AB} r_{AB}^* & \delta_{AB} r_{AB}^* & 0 & 0 & 0 \\ 0 & 0 & r_{AB}^* & 0 & 0 & r_{AB} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{AB}^* & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{AB}^* & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{BC}^* & \delta_{CB} r_{BC}^* & \delta_{BC} r_{BC}^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{BC}^* & r_{BC} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{AB}^* & r_{AB}^* & r_{AB}^* & r_{AB}^* \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma^* & 0 & \gamma \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{AB}^* & r_{AB} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$t \in \mathcal{C}$ について

$$P_t = \begin{pmatrix} r_{AB}^* & \delta_{CB} r_{AB}^* & \delta_{BC} r_{AB}^* & \delta_{BA} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{AB} r_{AB}^* & \delta_{AB} \delta_{BC} r_{AB}^* & \delta_{AB} \delta_{BC} r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & r_{AB}^* & 0 & \delta_{BA} r_{AB}^* & \delta_{BA} r_{AB}^* & 0 & \delta_{AB} r_{AB}^* & \delta_{AB} r_{AB}^* & 0 & 0 & 0 \\ r_{BC}^* & 0 & r_{AB}^* & 0 & 0 & r_{AB}^* & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & 0 & 0 & r_{AB}^* & \delta_{CB} r_{AB}^* & \delta_{BC} r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & r_{AB}^* & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & r_{AB}^* & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 & 0 & r_{AB}^* \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$t \in \mathcal{D}$ について

$$P_t = \begin{pmatrix} r_{AB}^* & \delta_{CB} r_{AB}^* & \delta_{BC} r_{AB}^* & \delta_{BA} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{BA} \delta_{CB} r_{AB}^* & \delta_{AB} r_{AB}^* & \delta_{AB} \delta_{BC} r_{AB}^* & \delta_{AB} \delta_{BC} r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & r_{AB}^* & 0 & \delta_{BA} r_{AB}^* & \delta_{BA} r_{AB}^* & 0 & \delta_{AB} r_{AB}^* & \delta_{AB} r_{AB}^* & 0 & 0 & 0 \\ r_{BC}^* & 0 & r_{AB}^* & 0 & 0 & r_{AB}^* & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & 0 & 0 & r_{AB}^* & \delta_{CB} r_{AB}^* & \delta_{BC} r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & r_{AB}^* & r_{AB}^* & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & r_{AB}^* & r_{AB}^* & 0 \\ r_{AB}^* & 0 & 0 & 0 & r_{AB}^* & 0 & r_{AB}^* & 0 & 0 & r_{AB}^* & r_{AB}^* & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

(平成 20 年 4 月 17 日受付)
 (平成 20 年 6 月 8 日再受付)
 (平成 20 年 6 月 19 日採録)



川西 俊之

1980 年生まれ。2008 年神奈川大学大学院理学研究科情報科学専攻博士後期課程修了。博士（理学）。コンピュータネットワークの性能評価理論に関する研究に従事。株式会社情報工房所属。



紀 一誠（正会員）

1968 年東京大学工学部計数工学科卒業。同年日本電気（株）入社。システム事業グループに 12 年間，研究開発グループに 19 年間在籍。この間コンピュータシステムの性能評価技術の研究開発とその応用に従事。1999 年 4 月より神奈川大学理学部情報科学科教授（工学博士）。