MANET上での効率良いビデオ配信を目的とした準最適マルチキャスト配送木の分散構築法

高島栄一[†] 村田佳洋[†] 柴田直樹[††]
安本慶一[†] 伊藤実[†]

モバイルアドホックネットワーク（MANET）上でビデオなどのマルチメディアデータをストリーミング配信するため，遅延や帯域幅などの複数のQoSに関する制約を満たすマルチキャスト木を動的に構築する手法が研究されている．MANET上のマルチキャスト木構築は，QoSの制約だけでなく，経路の安定性，省電力など，様々な目的に対する最適性を同時に考慮できることが望ましい．最適なマルチキャスト木を求める問題はNP困難であるため，近似解法として遺伝的アルゴリズム（GA）を用いる方法が提案されている．しかし，既存手法は，集中制御方式に基づいており，移動ノードの計算資源および通信資源の点で，大規模なMANETに適用するのは困難であった．本論文では，省電力性や通信の安定性など任意に指定した目的に対し，準最適なマルチキャスト木をMANET上の複数のノードで分散して計算し構築するGAベースの手法を提案する．提案手法では，MANETを複数のクラスタに分割し，各クラスタ内およびクラスタ間の2階層でそれぞれ配送経路を算出し，それらを組み合わせてマルチキャスト木とすることにより，スケーラビィリティを高める工夫を行っている．提案手法をネットワークシミュレータ上に実装し，既存手法の1つであるAQM（Ad Hoc Quality of Service Multicast Routing）と比較し，提案手法の優位性を示す．

A Method for Distributed Computaion of Semi-optimal Multicast Tree for Efficient Video Distribution in MANET

Eiichi Takashima,[†] Yoshihiro Murata,[†] Naoki Shibata,[††]
Keiichi Yasumoto[†] and Minoru Ito[†]

In order to realize multi-media streaming in mobile ad hoc network (MANET), a number of studies on dynamically constructing a multicast tree which satisfies multiple QoS restrictions such as bandwidth and delay, have been proposed. For construction of a multicast tree on MANET, it is desirable to take into account optimality of the tree in terms of communication stability, power consumption and so on. The problem to calculate an optimal multicast tree is known to be NP-hard. So, some existing studies propose algorithms based on genetic algorithms (GAs) to find a semi-optimal tree in practical time. However, since these existing methods adopt centralized mechanisms, they cannot be applied to large MANETs due to both communication and computation costs. In this paper, we propose a new method for MANET to dynamically construct a semi-optimal multicast tree which satisfies given QoS restrictions, for a given objective (e.g., communication stability and power consumption), by utilizing distributed computation of the tree based on GAs. In order to increase scalability, our proposed method constructs multiple clusters in MANET, and calculates a tree spanning all clusters and paths spanning nodes in each cluster by executing GA in some nodes selected in MANET. A multicast tree is constructed by grafting paths of all clusters on inter-cluster tree. Through experiments using network simulator, we confirmed that our method outperforms AQM (Ad Hoc Quality of Service Multicast Routing) in some objectives.

1. はじめに

近年，複数の移動端末が無線LANなどの近距離無線通信を用いて互いにパケットを交換し，目的ノードへパケットを中継することで，固定インフラを用いることなく広いエリアでのデータ通信を実現するモバイルアドホックネットワーク（以下，MANET）に関す

† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
†† 滋賀大学情報管理学科
Department of Information Processing and Management, Shiga University

る研究がさかんに行われている．また，MANET をITS，被災地での救援活動，街中での広告配信など様々な分野へ応用する試みが行われている．MANET において，動画などのマルチメディアデータをストリーミング配信することを目的として，ソースノードと複数の受信ノードを接続するマルチキャスト木を動的に構築する手法の研究がなされている．マルチキャスト木の構築においては，帯域や遅延などのマルチメディアデータの配送に特有の制約に加え，中継にかかるコスト（木の中間ノードにおける消費電力の総和など）やビデオ配信サービスを受けるユーザの数を最適化できることが望ましい．文献 1) では，マルチキャスト木におけるリンク数の総和を最小化する手法が提案されているが，遅延，帯域などの QoS に関する制約は考慮していない．一般に，複数の QoS に関する制約を同時に満たし，かつ与えられた目的に対し最適となるマルチキャスト木を算出する問題は NP 困難であることが知られている．MANET では，ノードの移動にともないリンクが生成・消滅しトポロジが動的に変化するため，トポロジの変化にあわせマルチキャスト木を再構築する必要がある．また，MANET で用いられる移動端末の計算能力，通信能力は限られていることも考慮する必要がある．以上より，MANET 上でビデオ配信のための最適なマルチキャスト木を構築・維持する問題は非常に難しいといえる．

準最適なマルチキャスト木を算出する問題に対する近似解法の 1 つとして遺伝的アルゴリズム（GA）を用いた方法が提案されている[2),3)]．しかし，これらの既存手法は集中制御方式を用いており，MANET 全体のトポロジ情報を 1 カ所に集め計算結果を配布するための通信コストと，木を算出する計算コストが大きい．したがって，MANET のノード数が多くなると，計算資源に乏しい移動ノード上で木の計算を行わせることは困難になる．

本論文では，MANET 上で，複数の QoS に対する制約を満たす準最適なマルチキャスト配送木を構築するための新しい手法（Hierarchical QoS Multicast Routing Using GA in MANET，以下 HQMGA）を提案する．HQMGA は，任意の指定された評価基準（消費電力最小，木の安定性最大など）を考慮することができ，ノード数に対するスケーラビリティを持つ．HQMGA では，多数の移動ノードからなるマルチキャスト配送木構築のためのコストを各移動ノードの計算・通信能力で扱える程度に抑えるため，MANET を複数のクラスタに分割し，すべてのクラスタをつなぐクラスタ間配送木，各クラスタにおける配送木を並列に別々のノードで算出する．そして，それらを組み合わせて 1 つの準最適なマルチキャスト木を求める．その際，クラスタをつなぐトポロジ情報を抽象化し，クラスタ間の配送木構築のために必要な計算・通信量をクラスタ内配送木と同程度まで削減する．以上を実現するため，GA による配送木の算出アルゴリズムに加え，クラスタ内のトポロジ情報の収集，クラスタ間でのトポロジ情報の収集，計算結果の移動ノードへの配布，ノード間での同期といった一連のプロトコルを提案する．

いくつかのクラスタサイズについて，HQMGA に基づいてマルチキャスト木を構築し，必要な計算量，通信量を調べた結果，MANET を適切なクラスタサイズに分割することにより，HQMGA はノート PC 程度の計算・通信能力で動作できることを確認した．さらに，HQMGA の有効性を検証するため，ネットワークシミュレータ GTNetS[4)] 上に実装を行い，QoS Multicast Routing の 1 つである AQM（Ad Hoc Quality of Service Multicast Routing[5)]）と性能を比較した．その結果，HQMGA は，消費電力最小，木の安定性最大など様々な目的に関して，より優れたマルチキャスト木を構築できることが分かった．

2. 関連研究

MANET 上でマルチメディア通信を実現するため，様々な QoS ルーティング手法が提案されている．

単一の QoS の制約条件を扱う手法が，文献 6)〜10) で提案されている．文献 6) は，指定した帯域幅（または遅延）を満たす経路を発見するための QoS ユニキャストルーティングの手法である．ある一定数の論理チケットを含むルートリクエストメッセージにより限定的フラッディングを行い，制約を満たす経路が存在するか調べる．メッセージを転送する際に，チケットを分配することで，フラッディングの範囲を限定している．文献 7), 8) は，指定した帯域幅を満たす経路を発見するための QoS ユニキャストルーティングの手法である．前者は，プロアクティブ型で，後者は，AODV に基づくオンデマンド型を採用している．これらの手法では，TDMA を用いることで，利用可能な帯域幅をタイムスロット単位に分けている．そのうえで，文献 7) では，未使用のタイムスロットから，制約を満たすまでタイムスロットの割当てを行う．一方，文献 8) では，AODV におけるルートリクエストの際に，ホップごとの利用可能な帯域幅を計測してメッセージに付加し，宛先のノードでは，指定した帯域幅に相当するタイムスロットの空きがある経路から来たメッセージ

にのみ応答を返す．ノードの集合をいくつかの領域に分割し，OSPF のように分散して QoS 制約を満たす配送経路を算出する手法として，CEDAR[9] が提案されている．CEDAR は，帯域幅に対する QoS のユニキャストルーティング手法であり，ネットワークのコアとなるグラフを動的に構築・維持する．ソースノードと目的ノードは，このコアとなるグラフを経由することで，必要な帯域幅を持つ経路を効率良く算出することができる．文献 10) は，複数の経路を同時に利用する QoS マルチパスルーティングの方法である．この手法では，まず，文献 6) と同様に論理チケットを使って，限定的フラッディングを行い，様々な経路上での利用可能帯域を調べる．利用可能帯域が分かっている複数の経路を組み合わせることで，要求された帯域幅を確保する．

帯域幅と遅延など複数の QoS に関する制約条件を同時に扱うルーティング手法として，以下のものが提案されている．文献 2) は，MANET 上の GA を用いたユニキャストルーティングの手法である．この手法では，トポロジ情報を抽象化することで，マルチキャスト木を高速に算出することができる．文献 3) は，無線環境下は不安定なネットワークであることを考慮し，QoS のメトリックを確率的に満たすマルチキャスト木を算出する手法を提案している．

MANET 上での QoS マルチキャストルーティング手法がいくつか提案されている．MCEDAR[11] は，CEDAR をマルチキャストルーティングに対応させたものである．CEDAR では，コアとなる経路を用意したが，MCEDAR では，コアとなるメッシュ型のマルチキャストグループを用意し，このマルチキャストグループを経由してデータを伝播させることで，効率良く配送経路を算出している．AQM[5] は，帯域幅に対するマルチキャストルーティングの手法である．要求–応答–予約の 3 つのフェーズにより，オンデマンドで必要な帯域幅を持つ経路を確保する．各ノードは，一定時間ごとに帯域幅の予約情報を隣接するノードと交換する．新たに経路を確保する際には，ルートリクエストを受け取った各ノードが，この情報から未使用の利用可能帯域幅を計算できるので，それをもとに必要帯域幅を持つ経路を確保する．

以上述べたように，MANET 上での QoS を考慮した，様々な経路発見手法が提案されている．しかし，我々が目標とする，複数の QoS 制約を同時に満たし，かつ，指定した目的（消費電力や安定性，サービスユーザ数など）に関して準最適なマルチキャスト木を分散制御により算出する方法については，既存手法では実現されていない．

3. 提案手法

この章では，まず提案手法の概要について述べ，次に提案手法で扱う問題を定義する．最後に提案手法の詳細を説明する．

3.1 概　　要

本論文では，MANET 上で，帯域と遅延に関する制約を満たし，指定した目的に関して準最適となるマルチキャスト木を構築する手法を提案する．

ノードの移動速度は，歩行者程度（4 km/h 程度），アプリケーションとして，動画や音声のストリーミング配信を想定する．各ノードはトランスポート層プロトコルとして，UDP などを使用し，再送は行わないものとする．各ノードは，マルチキャスト木の上流のノードからパケットを受信し，同じ内容のパケットを下流のノード群に対し送信するものとする．この際，1 ホップ以内の距離にある下流ノード群への送信は，上流ノードの 1 回のパケット送信により行うものとする．MAC 層プロトコルとして，IEEE802.11 ベースのプロトコルを想定し，キャリアセンスは行うが，RTS/CTS のやりとりは行わないものとする．また，各ノードは自身の移動速度，電波の到達範囲内にある他のノードとのおおまかな距離（電波強度などから推定する）を知ることができると仮定する．

MANET 上で，マルチキャスト配送木を構築する場合，木の計算を 1 つのノードで集中的に行おうとすると，必要なトポロジ情報をそのノードに集める必要がある．したがって，大規模な MANET では，計算量，通信量の観点から，集中制御で処理することは現実的ではない．提案手法では，MANET 上の複数のノードで，遺伝的アルゴリズム（GA）を使って配送木を分散して計算する．GA を用いるメリットは，MANET のトポロジの変化により配送木の再計算を行う際にも前回の計算に用いた解候補を使用することで，新しい木の算出が高速になること，任意の目的関数に対して，準最適な配送木を短時間で求めることが可能なことなどである．MANET では，ノードは移動するため，相対速度の大きなノードどうしを結ぶリンクは，切断されやすいと考えられる．提案手法では，マルチキャスト木全体の安定性を向上させるため，配送木上のリンクを選ぶ際，なるべく相対速度の遅いノードどうしのリンクを優先的に選ぶようにする．また，トポロジの変化に対応するため，配送木を定期的に再構築する．

配送木の計算の際に必要な情報収集の通信コストと計算量の両方を削減するため，提案手法では，OSPF

にならい，MANET を複数のクラスタに分割し，クラスタ内，クラスタ間のそれぞれで配送経路を計算し，組み合わせることで，マルチキャスト配送木を構築する．そのため，クラスタ内の局所的な配送経路の計算と，クラスタ間の大域的な配送経路の計算のそれぞれについて問題を定義する．

3.2 問題の定義

本節では，まず入力として与えられる MANET 環境について述べた後，提案手法が扱う大域的および局所的な配送木構築問題を定義する．

MANET のある時点でのトポロジを表す，重みつき無向グラフを $G=(V,E)$ とする．各ノードには固有の ID（整数値）が与えられるとする．2 つのノード v_i，$v_j \in V$ がお互いの電波の到達範囲内にあるときに限り，これらのノード間にリンクが存在すると考える．$s \in V$ をソースノード（マルチメディアデータの送信ノード）とする．マルチメディアデータの受信を要求するユーザノードの集合を $U \subset V$ と表記する．各ユーザノード $u \in U$ は，制約条件を表す 2 つの実数値（必要帯域幅 Br，許容遅延時間 Dr）を含むデータ配信要求 $req_u(Br, Dr)$ をブロードキャストするものとする．ここで，Br，Dr の値は，受信したいコンテンツによって一意に決まり，コンテンツごとに 1 つのマルチキャスト木を構築するものとする．ソースノード s と U に含まれるユーザノードを結ぶマルチキャスト配送木を $T=(V',E')$ と表記する（木には U のすべてのノードが含まれなくてもよい）．提案手法では，帯域幅，遅延時間に関する制約を満たし，かつ，与えられた評価関数 $f(T)$ を最大化するマルチキャスト木 T を求める問題を扱う．ただし，評価関数 $f(T)$ は，次節以降で述べるように，クラスタ間配送木，および，クラスタ内配送木で異なってよいものとする．以下では，ストリーミングサービスを受けることができるユーザノード数，通信の安定性，遅延の小ささ，の 3 つを同時に考慮した評価関数を用いて説明するが，提案手法では，任意の評価関数を設定して用いることができる．

3.2.1 クラスタ間配送木算出問題

本問題の目的は，以下の関数 $f_{\mathrm{global}}(T')$ で与えられる評価値を最大化するクラスタ間配送木 T' を求めることである（**図 1**）．ただし，図中のクラスタヘッドはクラスタを代表するノード，トップクラスタヘッドはすべてのクラスタを代表するノードであり，サブクラスタヘッドは，クラスタヘッドの処理を分散するためのノードである．GA の探索効率を向上させるために，各制約条件および配送木の不安定性についてのペナルティ関数（制約を満たさない場合に評価値を 0 とせず，幾分減じるための関数）を用いる．

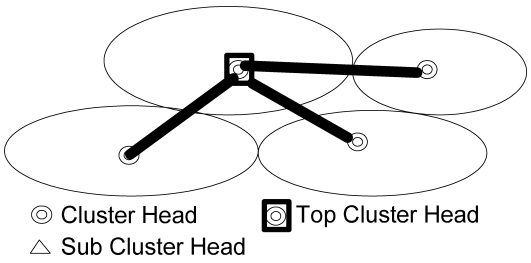

図 1 クラスタ間配送木
Fig. 1 Tree between clusters.

$$f_{\mathrm{global}}(T') = \alpha' \mathrm{F}^{\mathrm{U}}_{\mathrm{global}}(T') - \beta' \mathrm{F}^{\mathrm{D}}_{\mathrm{global}}(T') - \gamma' \mathrm{F}^{\mathrm{T}}_{\mathrm{global}}(T')$$

ただし，

$\mathrm{F}^{\mathrm{U}}_{\mathrm{global}}(T')$：遅延に関する制約条件を満たしたクラスタの数に関する関数
$\mathrm{F}^{\mathrm{D}}_{\mathrm{global}}(T')$：遅延に関するペナルティ関数
$\mathrm{F}^{\mathrm{T}}_{\mathrm{global}}(T')$：木の不安定性に関する関数

とする．α'，β'，γ' は正の重み付け係数である．

制約として与えられる，配送における遅延は，クラスタ間配送木の遅延とクラスタ内配送木の遅延の和であるが，この分配比率を δ（$0<\delta<1$）とおく．ただし，3.3.3 項で述べるように，帯域幅に関する制約条件は実装上の理由から設定していない．

$\mathrm{F}^{\mathrm{U}}_{\mathrm{global}}(T')$ は，以下の式で与えられる．

$$\mathrm{F}^{\mathrm{U}}_{\mathrm{global}}(T') = \sum_{C_i \in \mathscr{C}} \mathrm{deliver_c}(C_i, T')$$

ただし，

$$\mathrm{deliver_c}(C_i, T') = \begin{cases} 1 & \text{if } \mathrm{path_c}(C_i) \\ & \neq \emptyset \wedge\ \mathrm{Delay_c}(\mathrm{path_c}(C_i)) \leq \delta Dr, \\ 0 & \text{otherwise.} \end{cases}$$

$\mathscr{C}$ はネットワーク上にあるクラスタの集合，$\mathrm{path_c}(C_i)$ は，ソースノード s を含むクラスタのクラスタヘッド（クラスタの代表ノード，選出法は後述）からいくつかの中間ノードを経由し，クラスタ C_i のクラスタヘッドに至る経路のことである（もし，そのような経路が存在しない場合，$\mathrm{path_c}(C_i)=\emptyset$ とする）．また，$\mathrm{Delay_c}(path)$ は経路 $path$ におけるパケットの予想配送遅延（リンク遅延，ノードの処理遅延の総和）である．

$\mathrm{F}^{\mathrm{D}}_{\mathrm{global}}(T')$ は，遅延に関する関数であり，トップクラスタヘッド（すべてのクラスタのクラスタヘッドの

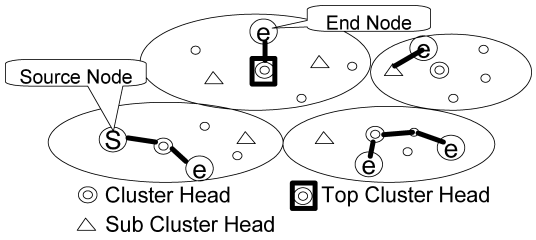

図 2 クラスタ内配送木
Fig. 2 Tree in a cluster.

中から選出されたノード．選出法は後述）から該当クラスタのクラスタヘッドまでのホップ数のうち，最大のホップ数について正規化した値を返す．以下の式で表される．

$$\mathrm{F}^{\mathrm{D}}_{\mathrm{global}}(T') = max_{C_i \in \mathscr{C}} \mathrm{hopcount}(C_i)/|V|$$

ただし，$\mathrm{hopcount}(C_i)$ は，s が所属するクラスタのクラスタヘッドからいくつかの中間ノードを通り，クラスタ C_i のクラスタヘッドに至る経路のホップ数を返す関数である（もし，そのような経路が存在しない場合，$\mathrm{hopcount}(C_i) = 0$ とする）．

$\mathrm{F}^{\mathrm{T}}_{\mathrm{global}}(T')$ は，木の安定性に関する関数であり，木 T' 上の各経路のノードの速度を，可能な最大速度 $SPEED$ について正規化した値を返す関数であり，以下の式で表される．

$$\mathrm{F}^{\mathrm{T}}_{\mathrm{global}}(T') = \{ max_{v \in \mathrm{path_c}(C_i)} \frac{\mathrm{speed}(v)}{SPEED} | C_i \in \mathscr{C} \}$$

3.2.2 クラスタ内配送木算出問題

本問題の目的は，以下の関数 $f_{\mathrm{local}}(T''_i, U_i)$ で与えられる評価値を最大化するクラスタ内配送木を求めることである（**図 2**）．

$$f_{\mathrm{local}}(T''_i, U_i) = \mathrm{F}^{\mathrm{U}}_{\mathrm{local}}(T''_i, U_i) - \epsilon'' \mathrm{F}^{\mathrm{T}}_{\mathrm{local}}(T''_i, U_i) \quad (1)$$

ただし，

$\mathrm{F}^{\mathrm{U}}_{\mathrm{local}}(T''_i, U_i)$：ユーザノードの数に関する関数
$\mathrm{F}^{\mathrm{T}}_{\mathrm{local}}(T''_i, U_i)$：木の不安定性に関する関数

である．

ここで T''_i，U_i はクラスタ内配送木，U_i はクラスタ C_i のユーザノードの集合である．また ϵ'' は重み付け係数である．

$\mathrm{F}^{\mathrm{U}}_{\mathrm{local}}(T''_i, U_i)$ は，木 T''_i 内において制約条件を満たしたユーザノードの数を返す関数である．制約条件は以下の式で定義される．

$$\mathrm{Band}(h_i, u, T''_i) \geq B_r$$
$$\mathrm{Delay}(h_i, u, T''_i) \leq (1-\delta) D_r$$

ここで $\mathrm{Band}(h_i, u, T''_i)$ は，配送木 T''_i 上のクラスタヘッド h_i からノード u までの経路における帯域幅の最小値であり，$\mathrm{Delay}(h_i, u, T''_i)$ は，同経路上の遅延時間の合計である．$(1-\delta)D_r$ はクラスタ内において許容される遅延の推測値である．

$\mathrm{F}^{\mathrm{T}}_{\mathrm{local}}(T''_i, U_i)$ は，木 T_i の不安定性に関する関数であり，以下の式で定義される．

$$\mathrm{F}^{\mathrm{T}}_{\mathrm{local}}(T''_i, U_i) = \max_{e \in T''_i}(\mathrm{instability}(e))$$
$$\mathrm{instability}(e) = \mathrm{distance}(v1(e), v2(e)) \cdot (\mathrm{speed}(v1(e)) + \mathrm{speed}(v2(e)))$$

ここで，$v1(e)$，$v2(e)$ は辺 e の両端点，$\mathrm{distance}(v_1, v_2)$ は頂点間の距離を返す関数，$\mathrm{speed}(v)$ は頂点 v の移動速度を返す関数である．

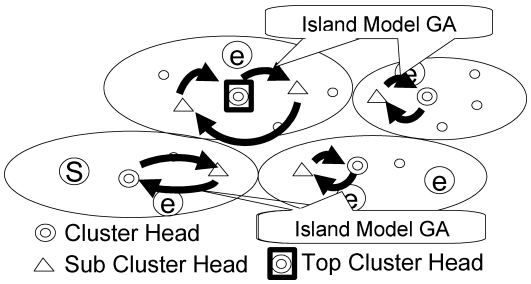

図 3 島モデル GA
Fig. 3 Island model GA.

3.3 提案手法の詳細

提案手法では，以下の 3 つのステップを繰り返す．

(1) ノードの集合をクラスタに分割する．
(2) クラスタどうしを結ぶバックボーンとなる木を算出する．
(3) 各クラスタ内の配送木を算出する．

全体の配送木は，クラスタ内配送木と，バックボーンとなるクラスタ間配送木を組み合わせることで算出する．各クラスタ内配送木の算出は，対応するクラスタ内で島モデル GA [12] を用いた並列分散処理（**図 3**）により算出する．

島モデル GA は並列 GA の一種であり，独立した解候補群を持ついくつかの GA を用いて解を求める手法である．隣接する解候補群と解候補を交換することにより協調計算を行う（図 3 の矢印）．島モデル GA を実行する計算機間の通信量が非常に少ない，多様性を保つ効果があるなどの利点を持つ．

ノードの移動にともなうトポロジの変化により，配送木の再構築が必要となる．提案手法では，上記の 3 つのステップを周期的に実行することにより，トポロジの変化に対応する．

上記ステップ 2 と 3 は，さらに，情報収集フェーズと配送木算出フェーズに分けられる．したがって，提案手法は，(A) **クラスタ分割**，(B-global) **クラスタ間情報**

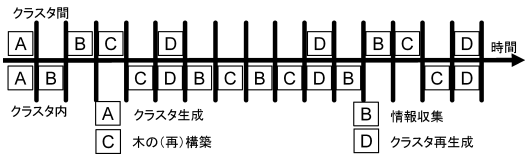

図 4 配送木の算出手順
Fig. 4 Computation procedure of multicast tree.

収集，(C-global) **クラスタ間配送木の算出**，(B-local) **クラスタ内情報収集**，(C-local) **クラスタ内配送木の算出**，(D) **クラスタ再構築**の 6 フェーズで行う．処理手順は**図 4** のとおりである．クラスタ内のトポロジの変化に比べ，クラスタ間のトポロジの変化の方が小さいと考えられるため，フェーズ (B-local) とフェーズ (C-local) は，他より高い頻度で繰り返す．

以下，各フェーズの詳細について順に述べる．

3.3.1 (A) クラスタ分割

このフェーズでは，ノードをクラスタに分割する．さらに，各クラスタごとに，クラスタヘッドとサブクラスタヘッドと呼ぶノードを選定する．クラスタヘッドとサブクラスタヘッド上で島モデル GA を実行する．また，クラスタヘッドの中から，最も小さな ID を持つノードをトップクラスタヘッドとして選定する．トップクラスタヘッドは，すべてのクラスタに関する情報を統括する．

MANET 上のクラスタの生成とクラスタヘッドの決定を行うアルゴリズムには，クラスタリングの再構成の負荷を軽減させた方式[13] や，ノードの ID を隣接するノードと $3D$ 回交換することでクラスタ半径が最大 D であるクラスタを構成する Max-Min D-Cluster [14] がある．

クラスタヘッドからの TTL= $D/2$ のフラッディングにより，各ノードで生成した乱数を収集し，上位 k 個のノードをサブクラスタヘッドとする．

3.3.2 (B-local) クラスタ内情報収集

このフェーズでは，各クラスタにおいて必要な情報をクラスタヘッドに集める．必要な情報は，以下の 5 点である．

- クラスタ内のノードとクラスタヘッド間の経路
- クラスタ内の接続を要求するノードの ID
- クラスタ内の各リンクの状態（未使用帯域，遅延，ノード間の推定距離）
- ノードの移動速度
- 隣接するクラスタの各組 (c_1, c_2) に対して，それらのクラスタヘッド間の経路

まず，各クラスタのクラスタヘッドより，クラスタ内の全ノードに対し，メッセージ（**図 5** の REQ Message $\mathbf{M_a}$）をフラッディングすることにより，クラスタ内の全ノードとクラスタヘッド間の経路を探索する．フラッディングされたメッセージを受け取ったノードは，クラスタヘッドに向かってリプライパケット（図 5 の REP Message $\mathbf{M_b}$）を送信する．リプライパケットは，経路を逆にたどってクラスタヘッドに届けられる．リプライパケットには，接続を要求するノードの ID と，リンクの状態の情報が含められる．

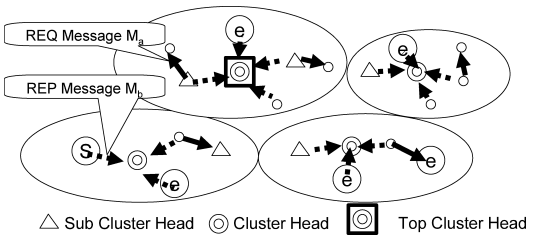

図 5 クラスタ内情報収集
Fig. 5 Gathering information in a cluster.

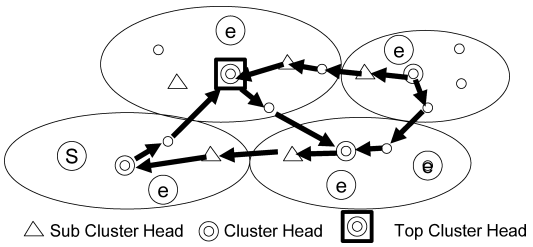

図 6 クラスタ間情報収集
Fig. 6 Gathering information between clusters.

各ノードが，隣接するクラスタより送信された，上記のフラッディングメッセージを受け取った場合，そのメッセージの送信元であるクラスタヘッドに対し，クラスタヘッド間の経路（以降，**隣接経路**と呼ぶ）の情報を送信する．

3.3.3 (B-global) クラスタ間情報収集

このフェーズでは，隣接するクラスタ間の経路（隣接経路）の帯域幅と遅延の情報を調べ，クラスタヘッドに集める（**図 6**）．また，これらの情報とユーザノードからブロードキャストされたデータ配信要求をクラスタヘッドからトップクラスタヘッドに集める．

隣接経路上のノードは，その隣接経路の遅延，帯域幅をクラスタヘッドに送る．隣接経路情報とクラスタ外のノードへのデータ配信要求を各クラスタヘッドからトップクラスタヘッドに集める．提案手法では，クラスタ間情報収集フェーズにおいて，クラスタヘッド間の経路における帯域幅と遅延時間に関する情報を調べて，トップクラスタヘッドに集める．この際，クラスタヘッド間経路のうち，必要な帯域幅の制約，遅延

時間の制約を満たさないものについては，トップクラスタヘッドに送信しない．そのため，送信された経路はすでに帯域幅の制約条件を満たしている．また，クラスタヘッドにおいて，クラスタ間配送木からの受信と，クラスタ間配送木における次ノードおよびクラスタ内配送木における次ノードへの配信は 1 回のブロードキャストにより実現している．以上のことから，クラスタ間配送木を計算する段階では帯域幅に関する制約条件を課していない．

3.3.4 (C-global) クラスタ間配送木の算出

このフェーズでは，トップクラスタヘッドが属するクラスタにおいて，島モデル GA を用いて 3.2.1 項で述べた問題を解き，クラスタ間配送木（図 1）の算出を行う．その後，算出結果は経路上の各クラスタへ送られる．このとき各クラスタヘッドは，送られた結果の中から 1 つ下流のクラスタの情報のみを保持する．

クラスタ間配送木は，ソースノードを含むクラスタのクラスタヘッドを送信ノードとし，各クラスタのクラスタヘッドを受信ノードとする配送木である．GA で用いる遺伝演算子については 3.4.1 項で述べる．

3.3.5 (C-local) クラスタ内配送木の算出

このフェーズでは各クラスタヘッドで 3.2.2 項で述べた問題を解き，クラスタ内配送木を算出する．クラスタ内配送木は，クラスタヘッド h_i を根とする配送木である．算出のために島モデル GA を用いる（図 3）．GA で用いる遺伝演算子については 3.4.2 項で述べる．算出結果は経路上の各ノードへ送られる．各ノードは，送られた結果の中から配送木上における 1 つ上流のノードと 1 つ下のノードの情報のみを保持する．

算出されたクラスタ内配送木の配送：

算出されたクラスタ内配送木に関する情報を，クラスタ内の関係ノードに配送する手法について述べる．まず，クラスタヘッドが送信パケットをブロードキャストする．パケットを受信したノードは，そのパケットの送信元を確認する．その送信元ノードが，クラスタ内配送木において上流のノードになっている場合にだけ，そのパケットをさらにブロードキャストする．この過程を繰り返すことでパケットを木の下流のノードへ伝達することができる．

3.3.6 (D) クラスタ再構築

フェーズ (D) では，ノードの移動によるトポロジの変化に対応するためのクラスタの再構成を行う．

新しく現れた，あるいは，所属していたクラスタに接続できなくなったノードは，接続要求メッセージを周囲のノードに流す．それを受け取ったノードは，自分の所属するクラスタのクラスタヘッドにメッセージを伝達する．

このメッセージを受け取ったクラスタは，フェーズ (A) と同様の方法を用いてクラスタの再構成を行う．

3.4 遺伝演算子

提案手法で用いる GA で採用した，配送木の符号化・復号化法，および遺伝演算子について述べる．GA を用いて木構造を最適化する場合，任意の木が生成可能な符号化方法を用いることがある．しかしこのような方法を用いた場合，解空間が広いために解の収束に大きな計算量が必要となる．逆に，解空間を狭くしすぎると，木の多様性を確保することが難しくなる．そこで提案手法では，両者の中間的な解空間を持つ符号化方法を考案した．この符号化法は，クラスタヘッド（あるいはトップクラスタ）から近いノード（クラスタ）を優先的に接続するため，ホップ数が比較的少なくなる点に特徴がある．

3.4.1 クラスタ間配送木用演算子

ソースノード s が属するクラスタをクラスタ 0 として，各クラスタに番号を振る．ネットワーク上のクラスタの総数を $\mathscr{C}$ とする．配送木の解候補（染色体）は，長さ $\mathscr{C}-1$ の数列 $< g_1, ..., g_i, ..., g_{\mathscr{C}-1} >$ とする．g_i には，クラスタ i が隣接するいずれかのクラスタの番号（隣接するクラスタがない場合 -1）が入る．

復号化の方法は以下のとおりである．クラスタ間配送木を $T' = (V', E')$ とする．

(1) T' の頂点集合 V' の初期値を空集合とする．s が属するクラスタを V' に加える．T' の辺集合 E' の初期値を空集合とする．
(2) j を 1 とする．flag を $false$ とする．
(3) g_j が -1 であればステップ (7) へ．そうでなければステップ (4) へ．
(4) クラスタ g_j が V' に含まれていればステップ (5) へ．そうでなければステップ (7) へ．
(5) クラスタ j が V' に含まれていなければステップ (6) へ．そうでなければステップ (7) へ．
(6) クラスタ j を V' に加える．また，クラスタ j とクラスタ g_j を結ぶ辺を E' に加える．flag を $true$ とする．
(7) j に 1 を加える．j が $\mathscr{C}-1$ 以下であればステップ (3) へ．そうでなければステップ (8) へ．
(8) flag が $false$ ならば，アルゴリズムを終了する．そうでなければステップ (2) へ．

クラスタ間配送木を求める GA では，交叉は一様交叉，突然変異は遺伝子座ごとにランダムな対立遺伝子に置き換える方法を用いる．

3.4.2 クラスタ内配送木用演算子

染色体表現，交叉方法，突然変異方法は，クラスタ間配送木の算出と同様の方法を用いる．ただし，配送木の解候補（染色体）は，長さの $|C|-1$ 数列 $< n_1, ..., n_i, ..., n_{|C|-1} >$ とする．$|C|$ は，あるクラスタ C のノードの総数とする．クラスタヘッドを 0 として，それ以外のノードにも一意に番号を付ける．n_i には，ノード i が隣接するいずれかのノードの番号か，木に接続しないことを意味する遺伝子（-1）が入る．

復号化の方法は以下のとおりである．クラスタ内配送木を $T''=(V'',E'')$ とする．

(1) T'' の頂点集合 V'' の初期値を空集合とする．クラスタヘッドのノードを V'' に加える．T'' の辺集合 E'' の初期値を空集合とする．
(2) j を 1 とする．flag を $false$ とする．
(3) n_j が -1 であれば，ステップ (7) へ．そうでなければステップ (4) へ．
(4) n_j が V'' に含まれていればステップ (5) へ．そうでなければステップ (7) へ．
(5) ノード j が V'' に含まれていなければステップ (6) へ．そうでなければステップ (7) へ．
(6) ノード j を V'' に加える．また，ノード j とノード n_j を結ぶ辺を E'' に加える．flag を $true$ とする．
(7) j に 1 を加える．j が $|C|-1$ 以下であればステップ (3) へ．そうでなければステップ (8) へ．
(8) flag が $false$ ならば，アルゴリズムを終了する．そうでなければステップ (2) へ．

4. 評価実験

提案手法の有効性を調べるために，(1) 配送木の算出時間，(2) 通信コスト，(3) 時間経過に対するパケット到達率の推移，についての実験を行った．

4.1 配送木の算出時間

提案手法のスケーラビリティを調べるために，ノード数の増加に対するクラスタ内配送木の算出時間を計測した．ここではすべてのノードが同じクラスタに属するよう設定している．移動端末上の動作を想定し，計算にはノート PC を用いた．実験環境は以下のとおりである：CPU Intel(R) Pentium(R) M processor 1,500 MHz，Windows XP，cygwin 1.5.18，gcc version 3.4.4.

実験結果を**表 1** に示す．この結果は 10 試行の平均値である．提案手法は，30 ノードを持つクラスタにおいても 1.5 秒程度で配送木を算出できることが分かった．クラスタ半径が最大 5 のクラスタリングを行ったときには各クラスタ内のノード数が 15 程度に抑えられたため，この算出時間は十分な短さであると考えられる．なお，クラスタ間配送木の計算時間は，ノード数が同じであれば，クラスタ内配送木の計算時間とほぼ同じである．

表 1 クラスタ内配送木の計算時間
Table 1 Computation time of tree in a cluster.

ノード数	計算時間（sec.）
10	0.140167
30	1.505700
50	5.498400
68	10.261367
100	26.932800
113	36.425200
225	164.233300
270	232.972300
450	902.678733

実験結果から，配送木の計算時間はノード数の 2 乗にほぼ比例する．配送木の計算時間はノード数（クラスタ数）が同じあればクラスタ内配送木，クラスタ間配送木であまり変わらないため，総ノード数 N に対してクラスタあたりのノード数が $\sqrt{N}$ のとき，マルチキャスト配送木全体の計算時間が最小になると考えられる．しかしクラスタ間配送木の算出の際に用いられる情報は抽象化の度合いが大きいため，総ノード数が少ない場合にもクラスタ内配送木の大きさはある程度必要であると考えられる．

4.2 通信コスト

提案手法の通信コストを調べるために，クラスタ内およびクラスタ間における配送木を 1 度構築するために必要なノードあたりの通信量をシミュレーションにより計測した．実験には，ネットワークシミュレータ GTNetS [4] を用い，実験空間は 3,000×3,000 m^2，ノード数は 1,000 とした．MAC 層プロトコルとして，IEEE802.11（最大伝送速度 2 Mbps）を用いた．無線の到達範囲を 160 m とした．本実験では，ノードは移動しないものとして計測した．クラスタリングには，Max-Min D-Cluster [14] アルゴリズムを用い，クラスタ半径を最大 3，5，10 と変え計測した．

実験結果を**表 2**，**表 3** に示す．これは 10 試行の計測結果の平均で，各フェーズにおける 1 ノードあたりの平均値とクラスタヘッドの 1 つあたりの平均値であり，送受信の通信量の合計である．表 2，3 から，クラスタリングの通信量が最も大きいことが分かる．クラスタリングの通信量やクラスタ間情報収集の通信量において，クラスタヘッドの通信量が 1 ノードあたり

表 2 1 ノードあたりの通信量（単位バイト）
Table 2 Required control traffic per node (Bytes).

クラスタ半径	3	5	10
クラスタ内の平均ノード数	7.7	15.2	36.0
クラスタリングの際の通信量	910.2	1,603.6	3,640.2
クラスタ間情報収集の通信量	341.8	291.6	259.8
クラスタ内情報収集の通信量	161.8	187.3	291.0
クラスタ間配送木配布の通信量	2.9	1.72	0.59
クラスタ内配送木配布の通信量	30.2	70.25	185.8
通信量の合計	1,446.9	2,154.5	4,377.4

表 3 クラスタヘッドの通信量（単位バイト）
Table 3 Communications traffic per one cluster head (Bytes).

クラスタ半径	3	5	10
クラスタ内の平均ノード数	7.7	15.2	36.0
クラスタリングの通信量	804.8	1,416.2	2,787.9
クラスタ間情報収集の通信量	302.8	236.7	185.6
クラスタ内情報収集の通信量	410.9	624.9	1,208.1
通信量の合計	1,518.5	2,277.8	4,181.6

の通信量より低い値となっているのは，隣接ノードがまったくないクラスタヘッドがいくつかあり，平均値を押し下げているためである．クラスタ内情報収集の通信量において，クラスタヘッドの通信量が 1 ノードあたりの通信量より高いのは，クラスタヘッドにクラスタ内のすべての情報が集まるためである．表 3 から，クラスタ半径が大きくなるにつれ通信量が増えているが，クラスタ半径が 10（クラスタ内平均ノード数 36）の場合でも，総通信量は 4.4 K バイトである．したがって，20 秒に 1 回木の再構築を行うことを想定した場合，必要通信速度は，通常ノードで 1.8 Kbps，クラスタヘッドで 1.7 Kbps となり，移動端末の通信量としては問題ない．一方，計算量に関しては，表 3 に示すように，クラスタ内のノード数（もしくはクラスタ数）が増えるにつれ，配送木計算の計算量は増大する．

したがって，N 個のノード数を持つ MANET では，通信量，計算量を考慮し，各クラスタが $\sqrt{N}$ に近い数のノードを含み，かつ，全体で $\sqrt{N}$ に近い数のクラスタを持つように MANET を分割することで，移動端末に必要な資源を最小化できる．

計算コストと通信コストの実験より，提案手法は，たとえば，ノード数 1,000 の MANET において，30 程度のノードからなる 30 個のクラスタに分割した場合，合計 3 秒程度の計算で配送木を算出することができる．またクラスタ内配送木を算出するための各クラスタヘッドの通信量は 4.2 K バイト程度に抑えることができる．これは，配送木の再構築が 20〜30 秒ごとに行われる場合には，ノート PC 程度の性能があり，IEEE802.11 b 準拠の無線通信が利用できる端末をクラスタヘッドに割り当てることにより，提案手法を現実の環境で実現可能であることを示している．

4.3 時間経過によるパケット到達率の推移

提案手法により構築された配送木を評価するために，配送木に含まれた（Join できた）ユーザノードにおける，経過時間に対するパケット到達率を計測した．これは受信できたパケット数をソースノードから送信されたパケット数で割ったもので，全 Join ユーザノードにおける値の平均値である．配送木構築過程およびその後のノードの移動によるトポロジの変化により，この値は時間とともに変化する．

この実験における設定は，通信コストの実験と同じ環境のものを用いた．ただし，ノードの移動速度が 0 km/h および 4 km/h の 2 つの場合について実験を行った．移動モデルはランダムウェイポイントを用いた．また，データ伝送速度は 64 Kbps とした☆．

また，提案手法が様々な QoS 指標を利用できることを示すため，配送木の評価関数は 3 章で記述したもの（木の安定性）（stable），3 章で記述したものから木の安定性を重視した項をのぞいたもの（non-stable），電力消費を最小化する項を加えたもの（power-saving），を用いて，配送木の時間経過に対するパケット到達率の変化を計測した．消費電力最小化手法として以下の設定を用いた．MANET 環境において携帯移動端末は基本的にバッテリ駆動であるため，消費電力を節約することは重要な課題である．実験において，端末のパケット送信時の消費電力は電波の送信距離の 2 乗に比例するとし，この消費電力量の総和を最小化する項を評価関数に追加した．算出されたクラスタ内配送木において，各ノードは必要な通信距離の 1.1 倍の距離まで届くよう電波強度を調整した☆☆．0.1 倍分はノード移動時の経路の安定性を考慮したためである．

提案手法のように任意の指定した評価基準に対し準最適なマルチキャスト木を構築するプロトコルは存在しないため，いくつかある QoS オンデマンドマルチキャストルーティングプロトコルの中で，比較的単純かつ性能に優れた AQM[5] を比較対象として選定した．AQM は周辺ノードの帯域幅の使用状況を追跡することにより，効率良く帯域幅を確保する手法である．構築された配送木の安定性を評価するために，配送木の再構築機能は用いなかった．ノードが移動しない場

☆ 最大伝送速度 2 Mbps の IEEE802.11 を用いているため，最大伝送速度 11 Mbps の IEEE802.11 b では，320 Kbps 相当となる．

☆☆ ノード間の距離は電波強度から推測できるものとしている．

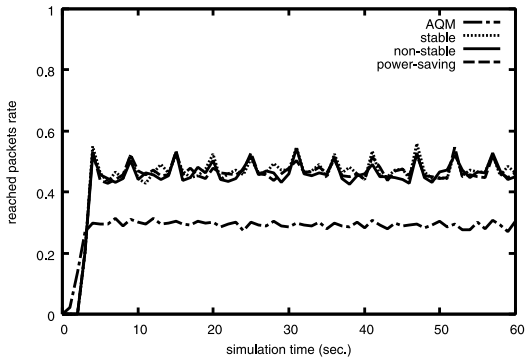

図 7 時間経過によるパケット到達率の推移：0 km/h
Fig. 7 Transition of the packet arrival rate over time: 0 km/h.

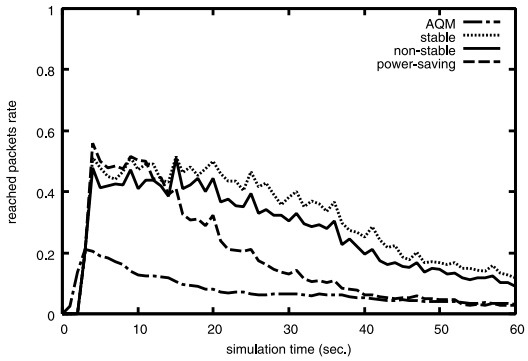

図 8 時間経過によるパケット到達率の推移：4 km/h
Fig. 8 Transition of the packet arrival rate over time: 4 km/h.

合とノードが歩行者程度の速度で移動する場合の実験結果をそれぞれ**図 7**，**図 8** に示す．

図 7 は，提案手法は AQM よりパケット到達率の点で優れていることを示している．また，ノードが移動しない場合，提案手法の 3 つの方法の間で大きな違いは見られない．一方，図 8 に示すように，ノードが移動するときは到達率が時間の経過とともに低下している．この低下の度合いは stable 法が最も緩やかであった．

次に，power-saving 法，stable 法，AQM それぞれに対し，通信のためにかかる消費電力をシミュレーションにより計測した．その結果，提案手法の stable 手法を 1 とした場合の消費電力量の比率は，power-saving 法が 0.61，AQM が 0.67 となり，power-saving 法の結果は，AQM や stable 法より少ない消費電力を示した．

以上のことから，提案手法は任意の指定した目的に最適化した配送木を算出する能力があると考えられる．

提案手法のデメリットとして，高い計算コストと即応性の不足があげられる．提案手法はクラスタヘッドとしてノート PC 程度の計算力を想定しており，PDA や携帯電話端末など，より計算能力が低い端末のみで MANET 環境が構成される場合には適用するのは困難であるかもしれない．また，配送木の安定性を考慮することができるが，いったん経路が切断されてしまえば再接続には配送木の再計算が必要となる．これは，オンデマンド型のルーティング手法とのハイブリッド手法を用いて改善できる可能性があると考えられる．

5. おわりに

本論文では，MANET 上でマルチメディアストリーミングを実現するための，複数の QoS 制約を満たし，かつ，与えられた目的に対し準最適なマルチキャスト木を，MANET 上の複数ノードで分散して構築する手法を提案した．提案手法の特徴は，通信の安定性，サービスを受信できるユーザ数，最大遅延などに対し，最適化したマルチキャスト木を，資源が限られた MANET 上のノードだけで構築可能なことである．シミュレーション実験により，提案手法が，既存のオンデマンド型のマルチキャスト木構築手法である AQM に比べ，ノード移動時の通信安定性に優れたマルチキャスト木を算出できることを示した．

今後の課題として，以下の 3 つがあげられる．第 1 に，提案手法で求めたマルチキャスト木の最適性の評価，実際の利用に際して有用となる様々な評価関数の考案である．第 2 に，配送木の最適性の改善である．現在のアルゴリズムではクラスタ内配送木とクラスタ間配送木の算出が独立して行われているが，情報を互いに利用することにより，より良い配送木の算出が可能だと考えられる．第 3 に，複数ストリームへの対応である．提案手法は単一のストリームを想定しているが，複数のストリームを流す場合においては，クラスタヘッドに負荷が集中すると考えられる．これを避けるために，ストリームごとにクラスタヘッドを変える手法を検討する．

参考文献

1) Mohapatra, P., Li, J. and Gui, C.: QoS in mobile ad hoc networks (2003).
2) Barolli, L., Koyama, A., Suganuma, T. and Shiratori, N.: GAMAN: A GA Based QoS Routing Method for Mobile Ad-hoc Networks, *Journal of Interconnection Networks* (*JOIN*), Vol.4, No.3, pp.251–270 (2003).
3) Layuan, L. and Chunlin, L.: QoS Multicast

Routing in Networks with Uncertain Parameters, *Web Technologies and Applications, 5th Asian-Pacific Web Conference* (*APWeb 2003*), pp.430–441 (2003).
4) Riley, G.F.: The Georgia Tech Network Simulator, *MoMeTools '03: Proc. ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, New York, NY, USA, pp.5–12, ACM Press (2003).
5) Bür, K. and Ersoy, C.: Ad Hoc Quality of Service Multicast Routing, *Computer Communications*, Vol.29, No.1, pp.136–148 (2005).
6) Chen, S. and Nahrstedt, K.: Distributed Quality-of-Service Routing in Ad-Hoc Networks, *IEEE Journal on Special Areas in Communications*, Vol.17, No.8, pp.1–18 (1999).
7) Lin, C.R. and Liu, J.-S.: QoS Routing in Ad Hoc Wireless Networks, *IEEE JSAC*, Vol.17, No.8 (1999).
8) Zhu, C. and Corson, M.S.: QoS Routing for Mobile Ad Hoc Networks (2002).
9) Sinha, P., Sivakumar, R. and Bharghavan, V.: CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm, *Proc. IEEE Conference on Computer Communications INFOCOM'99*, pp.202–209 (1999).
10) Liao, W.-H., Tseng, Y.-C., Wang, S.-L. and Sheu, J.-P.: A Multi-path QoS Routing Protocol in a Wireless Mobile ad Hoc Network, *ICN '01: Proc. 1st International Conference on Networking-Part 2*, London, UK, pp.158–167, Springer-Verlag (2001).
11) Sinha, P., Sivakumar, R. and Bharghavan, V.: MCEDAR: Multicast core extraction distributed ad-hoc routing, *Proc. Wireless Communications and Networking Conference* (1999).
12) Tanese, R.: Distributed Genetic Algorithms, *Proc. 3rd International Conference on Genetic Algorithms*, Schaffer, J.D. (Ed.), pp.434–439, Morgan Kaufmann Publishers (1989).
13) 谷口博人, 井上美智子, 増澤利光, 藤原秀雄：アドホックネットワークにおけるクラスタ構成法, 電子情報通信学会誌, Vol.J84-D-I, No.2, pp.127–135 (2001).
14) Amis, A.D., Prakash, R., Huynh, D. and Vuong, T.: Max-Min D-Cluster Formation in Wireless Ad Hoc Networks, *INFOCOM* (1), pp.32–41 (2000).

(平成 18 年 5 月 21 日受付)
(平成 18 年 11 月 2 日採録)

高島 栄一（学生会員）
2002 年創価大学工学部情報システム学科卒業．2004 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．現在，同大学院博士後期課程に在学中．

村田 佳洋（正会員）
2003 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．現在，奈良先端科学技術大学院大学情報科学研究科助手．遺伝的アルゴリズム，エージェント技術等の研究に従事．

柴田 直樹（正会員）
2001 年大阪大学大学院基礎工学研究科情報数理系専攻博士後期課程修了．現在，滋賀大学経済学部情報管理学科助教授．分散システム，ITS，遺伝的アルゴリズム等の研究に従事．IEEE/CS 会員．

安本 慶一（正会員）
1991 年大阪大学基礎工学部情報工学科卒業．1995 年同大学大学院博士後期課程退学後，滋賀大学経済学部助手．1997 年モントリオール大学客員研究員．2002 年より奈良先端科学技術大学院大学情報科学研究科助教授．博士（工学）．分散システム，マルチメディア通信システムに関する研究に従事．ACM，IEEE/CS 各会員．

伊藤 実（正会員）
1977 年，1979 年，1983 年にそれぞれ大阪大学基礎工学部卒業，基礎工学研究科博士前期課程修了，基礎工学研究科博士後期課程修了．1979 年より大阪大学基礎工学部助手．1986 年より大阪大学基礎工学部講師．1989 年より大阪大学基礎工学部助教授．1993 年 4 月より現在，奈良先端科学技術大学院大学情報科学研究科教授．関係データベース理論，オブジェクト指向のデータベースのアプリケーション，DNA プローブ等の研究に従事．ACM，IEEE 各会員．