

アドホックネットワークにおける End-to-End の帯域を考慮した アプリケーションレベルのルーティング方式

長田 俊明, 菅沼 拓夫, 白鳥 則郎
東北大学電気通信研究所/情報科学研究科

近年, 無線端末のみで構成するアドホックネットワークが注目されている. 一方, ネットワークの普及による利用用途の拡大とともに, マルチメディアアプリケーションを利用したサービスの需要が増加している. 我々は, マルチメディアアプリケーションのアドホックネットワーク上での効果的な利用を目指し, 通信アプリケーションに対し所望の end-to-end での通信品質を可能な限り提供する, アプリケーションレベルのルーティング方式を提案する.

Application Level Routing Scheme for End-to-End Bandwidth Provisioning in Ad Hoc Network

Toshiaki Osada, Takuo Sukanuma and Norio Shiratori
Research Institute of Electrical Communication / Graduate School of Information Science,
Tohoku University

Ad hoc network is receiving many attentions as a research topic for last few years. And demand for service of multi-media application is increasing along with expanding network use. We are aiming at effectively use multi-media application in ad hoc network. In this paper, we propose an application level routing scheme that provides the desirable end-to-end QoS for application as much as possible.

1 はじめに

無線ネットワーク環境において, 有線ネットワークに接続された基地局を中心としたインフラを使用せず, 無線端末のみでネットワーク構成する, アドホックネットワークが注目されている. 現在, アドホックネットワークのルーティング等に関する検討が, IETF の MANET ワーキンググループなどを中心に進められており, 様々なアーキテクチャやプロトコルが提案されている [1].

一方, ネットワークの普及による利用用途の拡大とともに, マルチメディアアプリケーションを利用したサービスの需要が増加している. しかし, アドホックネットワークにおいては, マルチメディアアプリケーションを考慮したネットワークサービスの実現は未だ検討段階である. これは, 従来のアドホックネットワークの研究においては, 安定した通信機能を提供するためのネットワークの接続性, および無線リンクの通信品質の向上に焦点が当てられていたためである.

本研究では, アドホックネットワークにおいて, 通信アプリケーションに対し, 所望の end-to-end での通信品質を可能な限り提供することを目指した, アプリケーションレベルのルーティング方式を提案する. 本方式は, 通信アプリケーションのフローに対して要求を与えたときに, 要求に合致した通信品質を実現するためにセッション単位に経路制御を行うものである. さらに, セッション単位で確立する経路上に通信品質 (帯域) を仮想

的に割り当てることで, トラフィック集中の回避をはかる. これにより, アドホックネットワーク内の複数セッションの各フロー品質要求を達成することができ, マルチメディアアプリケーションをアドホックネットワークにおいても効果的に利用できる.

本稿では, セッション毎の帯域割り当て情報を付加した経路選択手順, および, セッションの通信品質維持のための予備経路を利用した経路再探索手順について述べる.

2 関連研究

アドホックネットワークにおいては, DSR, AODV 等, 動的に変化するネットワークトポロジに対応した独自のルーティンプロトコルが提案されている [2][3].

しかし, アドホックネットワークにおけるこれらのルーティング方式は, 基本的には IP ルーティング技術に基づくものであり, 送信元と受信先の IP アドレスに対する経路を, ホップ数やリンク間の重みなどのメトリクスに基づいて確立する.

ここで, 音声や映像を扱うネットワークアプリケーションのような, 送信されるパケットに対してある程度の通信帯域が必要であるストリーミング型, 再生型のアプリケーションを利用することを考える. これらは, アプリケーションの各セッションに対して, 例えば, パケット損失無し且つ最大遅延値を保証, 僅少のパケット

損失と最大遅延値以下を保証，などの分類に基づいた提供すべきQoSがある。

しかしながら，各セッションに個別のQoSを要求している場合でも，IPパケットを処理単位として，単純なメトリックスに基づいたIPレベルの送信元-受信先の経路を使用するため，経路上でのセッションの重複等により必要とするリソースが不足し，それぞれのセッションのフローの要求品質が満たされない場合が生じる。

そこで我々は，アドホックネットワークにおいて，IPルーティングのように下位レイヤのネットワーク変化を追従するだけでなく，上位レイヤで利用するアプリケーションが要求するサービス品質(QoS)に対するフィードバックができる仕組みがあれば，マルチメディアアプリケーションをアドホックネットワーク上においても効果的に利用できるかと考えている。

3 アプリケーションレベルのルーティング方式

3.1 提案方式の概要

本研究では，2章で述べたアドホックネットワークにおけるIPルーティングの問題を考慮した，アプリケーションレベルのルーティング方式を提案する。

本方式の概要を図1に示す。

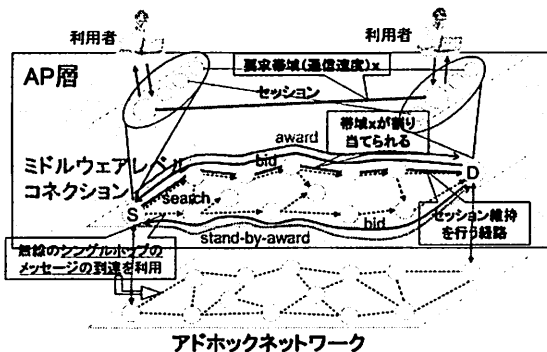


図1: アプリケーションレベルのルーティング方式

本方式は，(1)セッション単位のルーティングと，(2)アプリケーションレベルの制御，の2点を特徴とする。

(1)セッション単位のルーティング

セッション単位のルーティングとはすなわち，上位レイヤで動作している送信・受信アプリケーション間のセッション単位の経路を確立するものである。これにより各セッション特性に応じた制御が可能となる。

本方式では，セッション単位のルーティングに基づき，各フローに対してサービス品質維持のための資源確保が可能である経路を，動的に選択・変更する。

具体的には，アプリケーションのセッションに必要なとされるQoSとして，セッションを流れるフローに必要な下限の通信速度(帯域)に着目し，これを満たす経路を決定する。つまり，あるセッションに対して仮想的に割り当てが可能な帯域があるリンクを探索し，経路を

決定する。本稿で帯域に着目した理由としては，マルチメディアアプリケーションを使用するユーザが直接受けるサービスの品質，例えば動画の解像度やフレーム数等と，データの通信速度すなわち帯域が直接関連付くからである。

(2)アプリケーションレベルの制御

本方式では，セッション単位のルーティングをアプリケーション層内のミドルウェアにて行う。

アプリケーションレベルでの制御の特徴として，OS等のプラットフォームに依存せず導入が可能である点，上位レイヤのサービスと下位レイヤとの緩衝材として動作し，これにより上位層と下位層が効果的に連携して動作することが可能となる点などが挙げられる。

そこで，本方式ではIPルーティングのようにIPヘッダに経路情報を挿入するのではなく，ミドルウェアによってセッション単位の経路選択動作や，セッションと経路情報の管理を行う。

また，ネットワークレイヤ(IPレイヤ)に手を加えず，必要時にIPルーティングを利用することで，マルチメディアアプリケーションのように通信帯域の要求が必要でない環境や，要求を達成できる経路がない場合にもアプリケーションが停止しないような対処を行う。

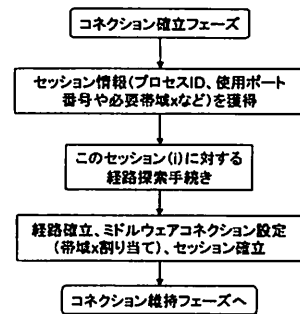


図2: コネクション確立フェーズ

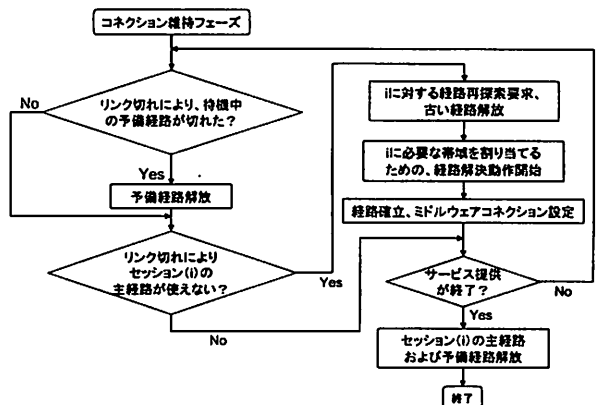


図3: コネクション維持フェーズ

本方式は，図2，3のように，大きく分けて，コネクション確立フェーズとコネクション維持フェーズから成る。コネクション確立フェーズは，アプリケーションからのサービス要求が発生した際に，要求に基づきミドル

ウェアレベルのコネクションを確立するための経路探索を行い、要求サービス品質を満たすセッションを確立するフェーズである。一方、コネクション維持フェーズは、トポロジの変化に追従して適切な経路を再探索し、動的に変更することによって、要求 QoS を満たしながらセッションを維持するフェーズである。

3.2 端末の保持する情報

本方式において、各端末は、C-info, C-route, および C-alloc という 3 つの情報テーブルを持ち、各テーブル内にはセッション識別子 APc に対してそれぞれ以下の属性 (要素) を持つ。

[C-info(APc) (各セッションに関する情報)]

(Src_gw, Dest_gw, Src_port, Dest_port, bw, state)

Src_gw : 要求発生元端末 (IP アドレス)

Dest_gw : 要求相手先端末 (IP アドレス)

Src_port : 要求発生元ポート番号

Dest_port : 要求相手先ポート番号

bw : セッションに対する割り当て帯域

state : セッション状態情報

[C-route(APc) (セッション毎の経路情報)]

(pre_gw, next_gw, flag)

pre_gw : 後ろ向きホップの端末 (IP アドレス)

next_gw : 前向きホップの端末 (IP アドレス)

flag : 経路状態

[C-alloc(APc) (隣接端末とのセッション割り当て対応状況)]

(gw, reserved_bw)

gw : 隣接端末 (IP アドレス)

reserved_bw : その隣接端末との割り当て帯域

C-route(APc) は、同じ APc であっても複数持つ場合もあり (予備経路情報として保持される場合である)、実際はそれらを区別している。

後ろ向きホップの端末とは、経路上で要求発生元端末の方向に 1 ホップ戻った端末を表す。前向きホップの端末とは、経路上で要求相手先端末の方向に 1 ホップ進んだ端末を表す。flag は主経路か予備経路かを識別するものであり、active なら主経路、stand-by なら予備経路である。

3.3 経路探索手続き

ここでは、図 2 のコネクション確立フェーズにおける経路探索手続きの部分の詳細を述べる。

(1) メッセージ形式の定義

経路探索には、search(l, D, bw, list, TTL, seq_num) というメッセージを使用する。ここで、D はセッションを確立するための相手先端末であり、bw は確保要求帯域を表す。TTL はメッセージの残りの生存ホップ数であり、要求発生元ではじめに指定した値から、search が端末を 1 ホップ伝播するごとに -1 される。list は要求発生元端末 S から現在の端末まで search が送られてきた経路のリストである。seq_num は、この帯域確保要求に対する経路

探索手続きの ID であり、他の seq_num と重複しないユニークな値を持つ。この seq_num には C-route 内の APc の値が入る。

n から m に返信される返答メッセージを bid(n, D, res_bw, res_list, list_length, seq_num) と表す。ここで、res_bw は、D から res_list を通って端末 m まで来た経路上で、アプリケーションのセッションに対して確保可能な帯域である。res_list は、search が S から D まで伝播したとき通った経路が格納される。list_length は、search が S から D まで伝播したとき通った経路の長さ (ホップ数) である。

bid に対する返答メッセージは、採用メッセージである award(bw, res_list, seq_num) と、予備経路として採用するメッセージ stand-by-award(res_list, seq_num) がある。

また、各メッセージの送信時には、返信されるメッセージ待ちに対するタイムアウトまでの時間を設定する。

経路探索におけるメッセージシーケンスを図 4 に示す。

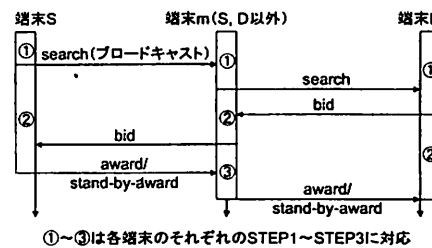


図 4: 経路探索におけるメッセージシーケンス

(2) 経路探索時の各端末での動作

(a) 要求元端末 S の動作

[step1 (search 送信)] 新しい C-info を獲得すると、S は search をブロードキャストし、bid を待つ。今、隣接端末を n とした場合、list=(S, n) である。

[step2 (bid 待ち)] 一つ以上の bid が返ってきた場合、bid を後述の (3) により評価して、選んだ bid の res_list 内で S の次に示される端末 n に、award を送信する。また、bid を返した他の端末の中で、任意の x 個の端末に stand-by-award(res_list, seq_num) を送信する。

これにより、res_list 内に含まれた端末を通る経路が、このセッションのために設定されるミドルウェアコネクションの経路となる。

このセッション i のための経路情報として、C-route テーブルに C-route(i) を加える。今の場合、C-route(i) は、主経路として (null, n, active) を設定し、予備経路として、(null, o, stand-by) を追加する。

また、C-alloc テーブルに C-alloc(i) として (m, bw) を追加する。

なお、タイムアウトまでに一つの bid も返ってこなければ探索手続きは失敗となる。

(b) 中継端末となる端末 m (但し、m ≠ S, D) の動作

[step1 (初期状態 (search 受信))] l から search を受信する。今受信した search 内の seq_num と同じ seq_num

を持った search を以前に受信していれば、search を廃棄し、m の処理は終了する。

$\sum \text{reserved_bw} + \text{bw} \times 2 > \text{Max_bw}(m)$ の場合、受信した search を廃棄し、m の手続きは終了する。ここで、 $\sum \text{reserved_bw}$ は、端末 m と m の全ての隣接端末との間に割り当てられているセッションの帯域の合計である。Max_bw(m) は、端末 m の無線ネットワークデバイスの最大帯域幅（通信速度）である。

TTL の値を 1 減らす。この時、TTL=0 になれば、search を廃棄し m での処理は終了する。

TTL ≠ 0 であれば、search をブロードキャストし、bid を待つ。ここで、list の最後には n が追加されており、list = (S,,,l,m,n) である。search 伝播規則を図 5 に示す。

[step2 (bid 待ち)] 複数の bid が返ってきた場合、bid を後に説明する (3) により評価・書き換えを行い、l に bid を返信して award/stand-by-award を待つ。bid 返信規則を図 6 に示す。

[step3 (award/stand-by-award 待ち)] award を受信すると、res_list の経路上で次に示される端末 n に対して award を送信し、C-route テーブルに C-route(i) として (l,n,active) を、C-alloc テーブルに C-alloc(i) として前向きホップ n と後ろ向きホップ l の (n,bw), (l,bw) を、それぞれ追加する。

stand-by-award を受信した場合、C-route テーブルに、予備経路として、(l,n,stand-by) を追加する。ただし、予備経路上の端末は、他の主経路上の端末とちがひ、仮予約という形をとり、C-route(APc) は (複数) 保持しているが、C-alloc(APc) を持たず、帯域を割り当てない。そのため、他のセッションに対する経路探索が発生した場合にこの分の帯域を割り当てることが可能である。この場合は、後述する予備経路切り替え手続き時に仮予約の解除を行う。

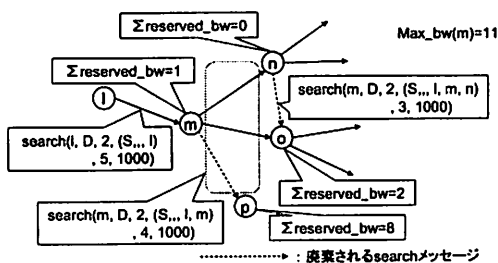


図 5: search 伝播規則

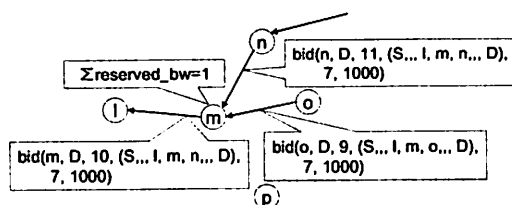


図 6: bid 返信規則

(c) 要求相手先端末 D の動作

[step1 (初期状態 (search 受信))] l から、search を受信する。

$\text{rest_bw}(D) = \text{Max_bw}(D) - \sum \text{reserved_bw}$ (この端末内で、割り当てられていない帯域) を計算する。rest_bw(D) > bw の場合は、要求相手がこのセッションのための要求帯域を確保できない問題があるが、この場合については検討中である。rest_bw(D) > bw の場合、list を res_list に、list の長さを list_length に入力する。また、res_bw = rest_bw(D) とし、l に bid を返信して、award/stand-by-award を待つ。複数の search を受信した場合、それぞれの search に対してこれらの処理を行う。

[step2 (award/stand-by-award 待ち)] l から award を受信すると、C-route テーブルに C-route(i) として (l,null,active) を追加する。また、C-alloc テーブルに C-alloc(i) として、(l,bw) を追加する。

stand-by-award を受信した場合、C-route(i) として、(l,null,stand-by) を追加する。ただし、C-alloc(i) は持たない。

(3) 中継端末での bid の評価・書き換え処理

[1] この端末に返信された複数の bid に対して、次の優先順位で一つの bid を選び、選んだ bid の list_length, res_list を返信 bid に使用する。

- list_length の値が最小の bid
- 最大の res_bw を返した bid
- ランダムに選んだ bid

要求発生元端末 S における bid の評価も、このようにして行い、その res_list を award や stan-by-award を返信する場合に使用する。

[2][1] で選んだ bid の res_bw と rest_bw(D) を比べて値が小さな方を返信する bid の res_bw に代入し、bid を search を送信してきた端末に返信する。

3.4 経路再探索手続き

ここでは、図 3 のコネクション維持フェーズにおける端末間のリンク切断が起こった場合の経路再探索要求手続き、再探索要求に基づく経路解決動作（予備経路を用いた経路探索）の詳細を述べる。

(1) 経路再探索要求手続き

(a) リンクが切れた端末 m の動作

C-alloc テーブルから、その隣接端末（今、n とする）とのリンクに帯域を割り当てられていたセッションを検索し、隣接端末 n とのリンクに帯域を割り当てられていたセッション全てに以下の処理を行う。

今、セッションを i とすると、C-route(i) を参照し、(l,n,active) を得たとする。

このとき、リンクが切れた側と逆方向のホップの端末に、要求発生元端末 S への再探索要求を伝えるメッセージである force-path-search(m,S,i) を送信する。C-alloc(i) と C-route(i) を削除する。

ここで、flagがstand-byである場合の処理は本稿では割愛する。

図7に、端末mと端末n間のリンクが消滅した場合の経路再探索要求動作を示す。

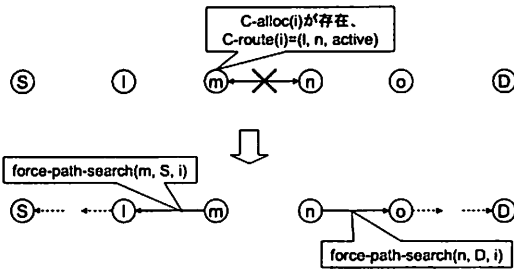


図7: リンク消滅による経路再探索要求動作

(b) 経路上の中継端末l (ただし、 $l \neq m, S$) の動作

端末kから、force-path-searchを受信すると、C-routeテーブルを参照して、セッションiに対するC-route(i)の内容、(u,k,active)より、端末kとは逆方向のホップである端末uへforce-path-searchを送信し、C-alloc(i)およびC-route(i)を削除する。

(c) 要求発生元端末Sの動作

端末kからforce-path-searchを受信すると、セッションiに対する経路解決動作(予備経路切り替え手続き)を開始する。経路解決動作が終了するまでの間は、セッションiに対してはIPルーティングに切り替わる。そのときは、C-info(i)のstateをoffにし、セッションiに対するフロー(データパケット)をIPレイヤに素通しする。

(2) 予備経路切り替えを用いた経路再探索

経路再探索には、以降で説明するような、予備経路上にしかメッセージを伝播させないような手続きにより、経路探索メッセージの氾濫を抑制する。

待機させていた予備経路に主経路としての使用開始要求を行うメッセージをreq-start(m, D, bw, list, seq_num)と表す。図9にreq-startの伝播規則を示す。req-startに対する予備経路として使用できなくなった場合の拒否メッセージをrejectと表す。また、(予備)経路解放指示のメッセージをreleaseと表す。

予備経路切り替えにおけるメッセージシーケンスを図8(a)に示す。図8(b)は、仮予約解除の場合のメッセージシーケンスである。

(a) 要求発生元端末Sの動作

[step1 (req-start送信)] C-routeテーブルを参照し、経路再探索が必要であったセッション(識別子をjとする)に対して、flag=stand-byであるC-route(j)(複数の場合もある)を探す。

この条件を満たすC-route(j)のnext_gw=nに対して、req-startを送信し、bid/rejectを待つ。今、list=(S)である。

C-route(j)が存在しない場合、1つも予備経路が残っていないので、通常の経路探索手続きを用いた経路再探索に移る。

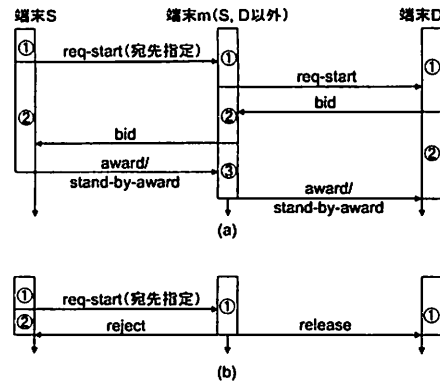


図8: 予備経路切り替えのメッセージシーケンス

[step2 (bid/reject待ち)] 端末nからrejectが返ってきた場合、この予備経路は既に帯域が割り当てられず(確保できず)に利用できないので、C-routeテーブルからC-route(j)に対する(null, n, stand-by)を削除する。

bidが返ってきた場合は、以降、通常の経路探索時の要求発生元Sのbid処理と同じである。そして、bid送信後、C-info(j)のstateをonにする。

bidが1つも返送されずにタイムアウトになった場合、通常の経路探索手続きを用いた経路再探索に移る。

(b) 中継端末m (但し、 $m \neq S$) の動作

[step1 (初期状態 (req-start/release受信))] req-start/releaseは次の送信先(IP)を指定して送信されるので、もしmが予備経路上の中継端末でない場合は、req-start/releaseというメッセージとして受信する前に廃棄される[終了]。

releaseを受信した場合、この予備経路(今、seq_num=j)に使う前向きホップの端末nにreleaseを送信し、C-route(j)を削除する[終了]。

req-startを受信した場合、 $\sum reserved_bw + bw \times 2$ を行う。これがMax_bw(m)より小さければであれば、前向きホップの端末nにreleaseを送信し、後ろ向きホップの(req-startを送信してきた)端末lにrejectを返信する。そして、C-route(j)を削除する[終了]。

$\sum reserved_bw + bw \times 2 > Max_bw(m)$ であれば、前向きホップである端末nにreq-startを送信し、bid/rejectを待つ。

[step2 (bid/reject待ち)] rejectが返ってきた場合、この予備経路は既に利用できないので、後ろ向きホップの端末lにrejectを返信する。そして、C-route(j)を削除する。

bidが返ってきた場合、通常の経路探索手続きに使うbid書き換え処理(3)を行い、後ろ向きホップ端末lにbidを返信し、award/stand-by-awardを待つ。このメッセージの受信処理は通常の経路探索手続きと同じ処理を行う。

4 評価

本方式の評価の方法として、実装ベースの評価とシミュレーションによる評価を行う予定である。

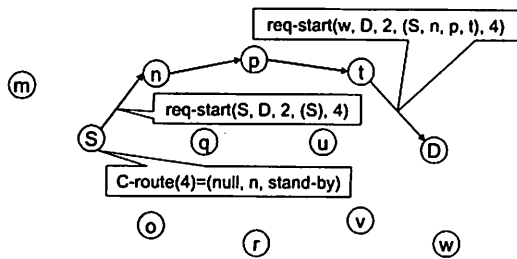


図 9: req-start の伝播

実装システムの構成を図 10 に示す。

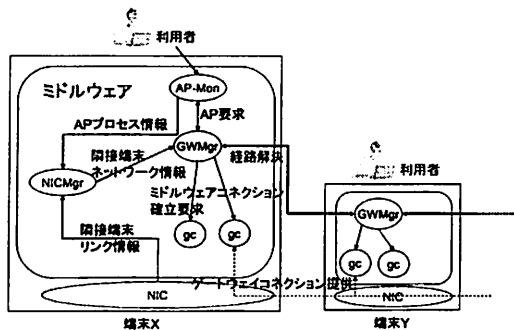


図 10: エージェントの構成

各端末内には、(a) アプリケーションの状況を獲得する機能、(b) ネットワークの状況を獲得する機能、および、(c) 他の端末と協調し、アプリケーションからのサービス要求に対する通信アプリケーション間のセッションを確立する機能、の 3 つの機能を構成する。これらの機能は以下のエージェントとして実現される。

AP-Mon: アプリケーションのセッション情報 (使用プロセス, 使用ポート, およびセッションを流れるフローに必要な帯域) を監視し, C-info や C-alloc の更新などに関与する (a)。

NICMgr: 無線ネットワークデバイス (NIC) を用いた, この端末と直接通信が可能である (電波が届く) 端末の検出, および AP-Mon から取得する情報により, C-info, C-route, C-alloc を保持し, これらの情報を経路解決動作の開始時に GWMgr に提供したり, 経路探索時に GWMgr からネットワーク状況の獲得要求に対して情報提供を行う (b)。

GWMgr: 他の端末の GWMgr と交渉し, 経路探索, 予備経路切り替え, 経路解放を行う。1セッションに対して, 1つの gc を生成する。この際, 自身の端末内のアプリケーションの状況, およびネットワーク状況を獲得する (c)。

ここで, gc は, GWMgr の経路解決の結果決定した他端末の gc との間に, ミドルウェアレベルでのシングルホップ・コネクションである, ゲートウェイコネクションを確立する。決定した経路に沿った各端末の対応する gc を直列に接続することで, そのセッションに対するミドルウェアコネクションを確立する。セッションのフローは, この連結したゲートウェイコネクションを順々

に通過して, フローの送信元のアプリケーションから最終受信先のアプリケーションまで渡される。

シミュレーションによる評価としては, 複数のセッションを時間をおいて開始して, 既存のアドホックネットワークにおけるルーティングプロトコルのみを用いた場合と本方式を利用した場合を比較し, 本方式を用いた方がそれぞれのセッションの要求する通信帯域を満足させることを示す。

5 まとめ

本研究では, アドホックネットワークにおいて, 通信アプリケーションに対し, 所望の end-to-end での通信品質を可能な限り提供することを目指した, アプリケーションレベルのルーティング方式を提案した。

本方式は, 通信アプリケーション間のセッション単位に確立する経路上に通信品質 (帯域) を仮想的に割り当てることで, アドホックネットワーク内の複数セッションの各フローの品質要求を可能な限り提供するものである。

また, 経路再探索においては, 予備経路上にしかメッセージを伝播させないような予備経路切り替え手続きにより, 経路探索メッセージの氾濫を抑制する仕組みを導入した。

本方式は, telnet や http 等のようなバースト的なデータ通信ではなく, マルチメディアアプリケーションやストリーミング等の様に定常的に帯域を消費する通信が行われる環境に特に有効なものであると考えている。

今後, 実装システムの動作検証および, シミュレーションによる評価とともに, 経路探索要求が同時に複数生じる場合の対処法や, 要求帯域を確保できる経路が発見できない場合等についても検討を進め, また, 端末のモビリティへ対応のために経路探索・再探索手法の改良を行う予定である。

参考文献

- [1] Internet Engineering Task Force (IETF) (<http://www.ietf.org>)
- [2] Johnson, D. and Maltz, D. "Dynamic source routing in ad hoc wireless networks," in Mobile Computing (1996).
- [3] Charles E. Perkins and Elizabeth M. Royer. "Ad hoc On-Demand Distance Vector Routing." Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.