

4-A モバイルプロトコル

座長 丹 康雄 (北陸先端大)

通信回線共有方式に適した再送制御

飯田 峰彦[†] 橋本 豊大[†] 富 強[†] 石原 進[‡] 水野 忠則[‡]

静岡大学大学院理工学研究科[†] 静岡大学情報学部[‡]

外出先から携帯端末を用いて無線でネットワークに接続するモバイルコンピューティングが普及しつつある。しかし、無線通信には帯域が狭い、信頼性が低いといった問題がある。そこで、筆者らは無線通信環境で論理的に帯域を広くし、より信頼性の高い通信を実現するための方式として通信回線共有方式を提案している。通信回線共有方式は携帯端末を複数集め、これらを相互接続してネットワーク(クラスタ)を構築する。各端末はクラスタのための回線とISP, LANなどの外部ネットワークへの接続回線を持つ。クラスタを構成する端末が外部ネットワークへ一つのデータを送受信する時に複数の外部通信経路を同時に利用、かつ共有することで、無線通信環境での通信速度の向上を図る方式である。筆者は本方式を実現するためのプロトコルとしてSHAKE(SHARed multilinK procedures for a cluster network Environment)を開発している。本稿ではSHAKEプロトタイプの問題について述べ、複数の通信回線を用いた、より効率的な再送制御について検討する。

Retransmission Control which is suitable for Sharing Multiple Paths System

Minehiko Iida[†], Toyohiro Hashimoto[†], Fu Qiang[†], Susumu Ishihara[‡], and Tadanori Mizuno[‡]

Mobile computing became popular. However, it has some problems such as narrow bandwidth, low reliability. To solve these problems, we have proposed "Sharing Multiple Paths System". In Sharing Multiple Paths System, some mobile hosts are connected each other and form a local network, which we call *cluster*. Each mobile host has a channel connected with other mobile hosts and a channel connected with outside network such as Internet or LAN. When a mobile host sends data to a destination in the outside network, data transmission speed becomes higher even in the wireless communication environments by using multiple paths at the same time and sharing it with some mobile hosts. We have developed the protocol *SHAKE* (SHARed multilinK procedures for a cluster network Environment) to realize the system. In this paper, we examine the problem by simulation. Furthermore we enumerated the problems of *SHAKE* prototype, examined more effective method for retransmission control.

1 はじめに

近年、ノートPC, PDAといった携帯端末や携帯電話, PHSといった無線通信インフラの急速な普及により、出張先や移動中の電車の中でもオフィスとほぼ同様の作業ができ、インターネットなどのネットワークへのアクセスを可能にするモバイルコンピューティングが普及している。しかしながら、無線通信環境では有線通信に比べて帯域が狭く、品質が変動するので信頼性が低い。また、大量のファイル、マルチメディアデータの転送には不向きである。

そこで、無線通信環境でも帯域を論理的に広く使い、高信頼な通信を実現するための方式として筆者らは通信回線共有方式[2]を提案している。また、本方式を実現するためのプロトコルSHAKE[3]を開発している。

IETF (Internet Engineering Task Force) のワーキンググループMANETではアドホックネットワークの

研究が行われている。MANETの研究目的は、トポロジの変化に伴う経路情報の更新といったインターネットで今まで使われてきたルーティングでは解決できない問題に対し、トポロジ変化に適応したルーティングアルゴリズムを開発することである。近年、アドホックネットワークのためのルーティングプロトコルが多数提案[7, 8]されており、それらの性能評価がシミュレーションによって行われている[9]。

アドホックネットワークと通信回線共有方式は、一時的に複数の端末を集めてネットワークを構築する分散協調型のネットワークに類する。2つのネットワークはアドホックネットワークが外部ネットワークへの接続回線は1つか全くないのに対し、通信回線共有方式はクラスタを形成する複数の端末が外部ネットワークへの接続回線を持っている点で異なる。

関連研究としては、通信する端末間にIPトンネリングを行う専用ゲートウェイを設置して、専用ゲートウェイが複数経路にパケットを振り分ける方式[4]がある。この方式を実現するためには専用のハードウェアを必要とし、専用ゲートウェイ間の複数経路は静的

[†] Graduate School of Science and Engineering, Shizuoka University

[‡] Faculty of Information, Shizuoka University

であるのに対し、通信回線共有方式ではパケットの振り分けは、通信を行う端末自身が行い、複数経路は動的に変化するという点で性質が異なる。

現在、通信回線共有方式はプロトタイプの実装によって転送速度が向上することを確認している。プロトタイプではクラスタ内通信およびクラスタと外部ネットワークとの通信にTCPを用い、その上位層で複数の回線を制御しているが、効率面から考えると複数のTCPコネクションを用いるよりもUDPの上位層にSHAKEを実装するか[1]、TCPもしくはIP層を通信回線共有方式に適したプロトコルに拡張した方がTCPのオーバーヘッドを減らすことができ、より柔軟な誤り制御、再送制御が可能となる。

そこで本稿では、TCP、IPを拡張するというアプローチのもとでより適切な再送制御について検討する。以下第2節では通信回線共有方式について述べ、第3節でプロトタイプの問題点について、第4節でプロトコルの拡張について述べ、第5節でまとめを述べる。

2 通信回線共有方式

2.1 通信回線共有方式の概要

通信回線共有方式は、インターネットなどのネットワークへの接続回線を持った携帯端末が複数集まり、相互接続して回線を共有しあうことで論理的に回線を太くして通信する方式である。この相互接続されたネットワークを我々はクラスタと呼んでいる。各端末は2つのネットワーク（クラスタと外部ネットワーク）に接続するため、ネットワークインターフェースを2つ持つ必要がある。また、外部ネットワークへ接続するためのアカウントをISP (Internet Service Provider) や研究機関を通じてユーザが用意することも必要である。このような環境を用意して本方式を利用すると、以下のような利点がある。

- 無線通信環境における転送レートの向上
- クラスタを構成する各モバイルホストは移動可能なので動的にクラスタを構成可能
- 外部通信路としての通信媒体は各モバイルホストが選択可能
- 外部通信路を持っていなくてもクラスタで接続された他のホストの回線を利用可能
- 複数の回線にパケットを分散させることで伝送誤りを通信路によって独立化できるので誤り訂正が容易

本方式は回線の共有を実現するためのデータ配信に特徴がある。1つのデータファイルを一つの回線を選択

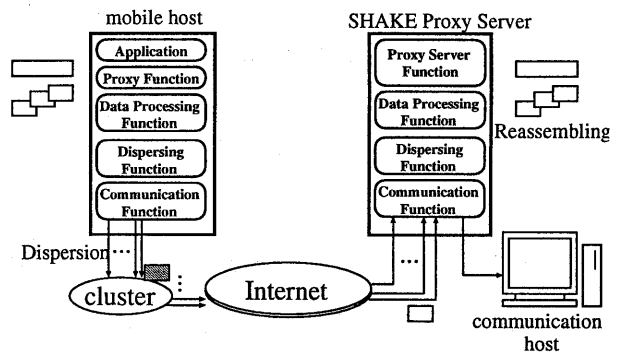


図1 SHAKEの機能構成

して送信するのではなく、ある一定サイズのパケットにファイルを分割して、各端末が持つ外部通信路すべてを同時に利用して並列に配送する。そのため、ファイル転送のためのオーバーヘッド（ファイル分割、振り分け、再整列）が従来の通信よりも増加する。

本方式においてクラスタ形成のための通信媒体の選択は重要である。パケットの振り分け、再整列などのオーバーヘッドはクラスタ内の転送速度に大きく左右されるからである。各端末が持つ外部通信路よりクラスタ内通信路の方が伝送遅延が大きい場合、パケットの振り分けに時間がかかってしまい、互いのもつ回線を有効に利用することができない。

どのような環境で本方式を利用するかによっても通信媒体は変わってくる。例えば室内で携帯端末を一時的にHubに接続するような環境であれば、転送レートが高い有線でクラスタを形成して外部回線を共有するのが無難である。また、災害などによって必要なハードウェアが用意できないような環境では、無線でクラスタを形成できると便利である。無線LANは基地局が用意できれば屋外であってもクラスタの形成は可能だろうし、携帯電話やPHSでも緊急時の場合なら十分事足りる。

2.2 通信回線共有プロトコル SHAKE

通信回線共有方式を実現するためのプロトコルとして、筆者らは通信回線共有プロトコル SHAKE (SHARed multilinK procedures for a cluster network Environment) を開発している [3]。SHAKEの機能構成を(図1)に示す。SHAKEはクラスタを形成するすべての携帯端末と、通信相手の固定ホストのトランスポート層(TCP)とアプリケーション層の間に組み込まれる。

SHAKEは、これらを実現するために以下の4つの機能を持つ。

- プロキシ通信機能
クライアントアプリケーションがSHAKEをプロキシとして設定することで任意のホストは

SHAKE を組み込むことなく SHAKE の機能を利用できる。

- 通信制御機能

クラスタおよび外部通信路の接続管理、経路情報の管理を行う。

- データ加工機能

データ送信時は SHAKE によって専用のヘッダが付加され、一定サイズの専用パケットに分割されて TCP に送られる。再整理時には宛先のホストが SHAKE データパケットを受信するとシーケンス番号を見てアプリケーションへ渡せるパケットであればそのまま渡し、そうでなければ一度バッファに入れられる。次にバッファを見てアプリケーションへ渡すべきパケットが存在すればそれを渡す。以上の処理を繰り返す。

- 振り分け機能

SHAKE パケットを自分が持つ外部通信路、およびクラスタ内ホストへ送信する。

3 プロトタイプの問題点

現時点での SHAKE プロトタイプは TCP とアプリケーションの間のプロトコルとして実装されている [3]。このため、通信の信頼性保証は TCP で行われる。通信回線共有方式の通信は複数の経路を使用するとはいえ、あくまで送信元 S と宛先 D の間の通信であり、S と D の間で接続を確立して仮想的に単一接続の信頼性を保証するのが理想である。しかしながら、プロトタイプの実装では送信ホストがパケットを振り分ける時、自分の持つ外部通信路を通るパケット以外は、クラスタ内の中継ホストを経由して中継ホストが持つ外部通信路を通過する。したがってパケットはクラスタ間をつなぐ TCP 接続およびクラスタと宛先ホストとをつなぐ TCP 接続の 2 つの TCP 接続を通過する。このため、パケット転送のための制御は TCP の上で行わなくてはならなくなり、オーバーヘッドが大きくなる。

プロトタイプでクラスタを n 台で形成したとき、TCP 接続は全体で $2n - 1$ 必要になる。TCP における信頼性の保証は TCP 接続単位で管理されるため、各接続での制御は完全に独立してしまっている。そのため、一つの経路を一つの接続と見なすことができない上に、エンドツーエンド間を一つの接続と見なすことができないため、複数経路の存在を積極的に活用した再送制御が困難である。

3.1 パケット再整理

プロトタイプのような接続環境ではクラスタ内の端末を接続する TCP 接続、およびクラスタと外部の間の TCP 接続でパケットの再整理が行われている。さらに、SHAKE 層で複数の TCP 接続から得られるパケットの再整理も行う必要がある。

3.2 再送

通信回線共有方式では、複数の回線を同時に利用しているため、パケット再送が必要になった場合には、必ずしも再送が必要になったパケットが以前通った経路で再送をする必要がない。すなわち、通信回線共有方式では再送が必要になったときは、より適切な経路を使って再送するというアプローチが可能である。しかしながら、一つの経路は二つの TCP 接続で接続されているため、ACK は直接送信ホストもしくは宛先ホストに届けられない上に、TCP の再送処理は接続毎に行われているため、ある接続でタイムアウトになった場合別の経路で再送できない。

4 プロトコルの拡張

前節で述べた問題から、TCP は通信回線共有方式の実現には不向きである。そこで、本方式に適したプロトコルについて検討してみる。

問題解決のアプローチとして、

- TCP の拡張

- IP の拡張

が考えられる。それぞれについて以下で詳しく検討する。

4.1 TCP の拡張

プロトコルは IP の上に新しいプロトコル (SHAKE) を設置する。SHAKE 層は信頼性のある通信を実現する層 (SHAKE A 層と呼ぶ) と、複数の経路を管理しルーティングを行う機能を提供する層 (SHAKE B 層と呼ぶ) の 2 階層で構成される。すなわち、SHAKE A 層は TCP/IP における TCP、SHAKE B 層は IP に相当する機能を担当する。SHAKE A 層は SHAKE B 層から取得した各経路に対応するパラメータを用いて再送タイムアウト値を算出する。そして SHAKE では、アプリケーションからはエンドツーエンド間の経路が一つであると見なすことができるようなインタフェースを想定する。クラスタ内の端末は、自分が送信を行わない時は送信ホストから送られてきたパケットを UDP

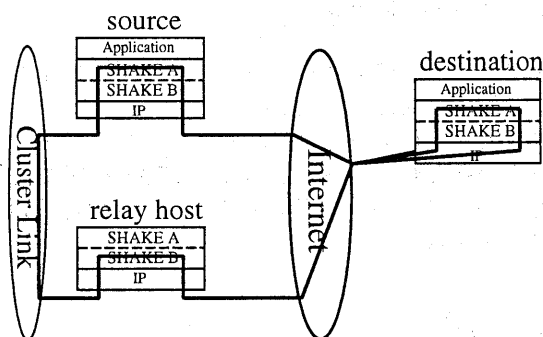


図 2 SHAKE プロトコルの関係

を介して宛先ホストにフォワードするルータとしての役割を果たす。

4.1.1 複数経路の管理

SHAKE B 層は複数経路を管理するテーブル（以後、経路情報管理テーブル）を持つ。すなわち、テーブルは一つの経路に対して、

- コネクション（経路）識別子
- IP アドレス
- 経路の状態を示すパラメータ (RTT など)

を持つ。

4.1.2 確認通知

TCP 標準の確認通知では受信側がセグメントを断続的に受信した場合、同じ ACK が複数回送信側に送られることになる。通信回線共有方式のような無線媒体の複数経路を同時に使用するような環境では

- 経路毎で転送速度が異なる
- パケットロス率が高い

といった理由から受信側がセグメントを断続的に受信する可能性が高くなり、その結果 ACK の重複が頻繁に発生するためスループットは著しく低下する。そこで、この問題を解決するために SACK オプション [6] を利用する。SACK はセグメントが断続的に受信された場合、受信側が送信する ACK にオプションとしてすでに受信したセグメントブロックの情報を付加して送るため、無駄な再送が減少する。

複数経路を利用することで、ACK の送信に利用する経路は柔軟に選択可能である。方法としては以下のものが考えられる。

- パケットが通って来た経路を使って ACK を送信する方法
- RTT が最短の経路に ACK を送信する方法

前者の方法を実現するためには、各セグメントがどの経路を通ってきたのかをセグメント受信側の SHAKE A 層で分かっている必要がある。しかしながら、再送制御に経路制御を組み合わせるのは TCP の実装から判断しても複雑になるのは必至である。しかも本稿で SHAKE A 層と B 層に階層化した理念にも反する。

後者の方法は、ACK が送信される経路が送信したセグメントが通過した経路に依存しないため、SHAKE A 層では経路情報に関知する必要がなくプロトコル階層化の理念に合致している。ただし、SHAKE A 層とは独立に SHAKE B 層が RTT を測定するための仕組みが必要になる。

以上の考察から、確認通知の方法としては RTT が最短の経路に ACK を送信する方法が適切であると考えられる。

4.1.3 再送制御

再送も確認通知と同様、複数経路をどう利用するかがスループットにも影響する。再送方式としては以下のものが考えられる。

- 最初と同じ経路を使用する方法

セグメントが最初に通過した経路を知るためには SHAKE B 層から情報を得なければならない。また、再送がほとんど起こらない経路と、同じパケットが複数回再送される経路とでスループットに大きな差が生じ、複数経路全体としてのスループットが低くなると予想される。

- 再送パケットを RTT が最も短い経路で送信する方法

RTT を経路選択の基準にする。送信側の SHAKE A 層は経路のことには関知する必要がなく、経路を選択するのはあくまで SHAKE B 層である。また、受信側の再整列処理の待ち時間を減らすことができ、パケット最整列の高速化が期待できる。ただし、確認通知の場合と同様、RTT を別途測定する必要がある。

再送されるパケットは信頼性が高く、転送速度が最速の経路で送信するのが望ましい。そのため、再送パケットは複数経路を同時に使用するよりも一つの経路で送るべきである。したがって、RTT が最短の経路で送信するのが適切であると考えられる。

TCP では再送タイムアウトを設定して、ACK がタイムアウト時間内に受信できない場合に再送を行う。現在多くの TCP 実装では再送タイムアウトは、RTT を用いて

$$RTT = \alpha RTT + (1 - \alpha)M \quad (1)$$

$$D = 7/8D + 1/8|RTT - M| \quad (2)$$

$$Timeout = RTT + 4D \quad (3)$$

で算出される。(なお、M:RTT の実測値)これに対し、通信回線共有方式は複数の経路を同時に管理するため、TCP よりもタイムアウトの設定は困難である。SHAKE A 層は経路情報には関知せず、RTT の測定は SHAK E B 層で行い結果を SHAK E A 層が受け取り再送タイムアウトを算出するのが望ましいと考えられる。

そこで、再送タイムアウトの設定方法としては SHAK E A 層が B 層からの RTT、および経路の重みを取得し、式 (3) に経路の重みをかけたものをタイムアウト値とする方法が考えられる。

4.1.4 輻輳制御

既存の TCP で実装されている輻輳制御は、送信側の輻輳ウィンドウと受信側の受信ウィンドウに従って、送信側が一度に送信できるセグメントのバイト数を制限している。ウィンドウ幅は送信セグメントに対する ACK がタイムアウト以内に送信側に返って来ると、ウィンドウ幅は増加し、受信ウィンドウに達するとタイムアウトが発生するまで一定値を保持する (図 3)。しかしながら、タイムアウトが発生すると、ウィンドウ幅は最大セグメントサイズにまで絞られてしまう (図 3 の斜線部)。無線通信環境では帯域が狭い上にパケットロス率が有線通信環境に比べて高い。したがって、タイムアウトが頻繁に発生することが予想されウィンドウ幅の急激な変化が起こる回数も多くなる。その結果、送信セグメントのサイズは常に低い値に設定されるために狭い帯域を有効利用できない。

[5] で提案されている WTCP では TCP の再送タイムアウト、ウィンドウを用いずに送信側と受信側で送受信レートを測定し、互いに測定したレートをデータパケットもしくは ACK に含めて送信する。受信側は短い時間単位で受信レート (savg_ratio) を、それよりも長い時間単位で求めた受信レート (lavg_ratio) を測定し、それぞれの値を閾値と比較し、その大小関係に応じて送信側の送信レートを増減させる。送信レートの減少率は状況によって変化する。送信レートが閾値

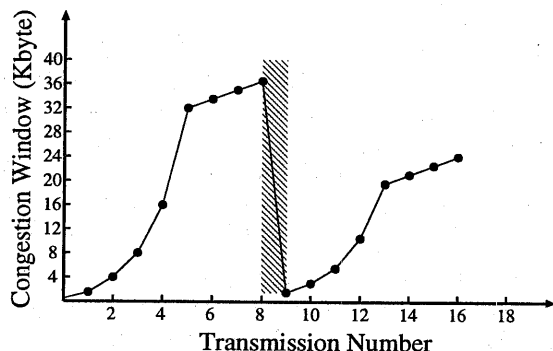


図 3 TCP の輻輳アルゴリズム

以下になった場合は減少率は 20 %、受信側が断続的なセグメントを受信した場合は 50 % になる。減少率は送信レートが増加すると 20 % にリセットされ、減少を続けると増加する。WTCP は TCP のように送信レートを減少させるときに最小値にするのではなく、緩やかに減少させるところに特徴がある。WTCP のレート制御を用いると、TCP Reno では帯域利用率が 75 % であるところが 90 % まで増加させることができる。

通信回線共有方式では複数の経路を利用する。トランスポート層では経路情報に関知することなく、信頼性を保証するのが望ましい。しかしながら、各ホストが持っている外部通信路の無線媒体はそれぞれ異なっており、帯域、伝送遅延も経路によって異なるため、WTCP のように無線通信環境に適した、複数の通信帯域を有効利用できるようなプロトコルが必要である。そこで考えられる方法として、経路情報に依存する制御と依存しない制御手法がある。

前者は、SHAK E B 層で輻輳制御を行う。各経路で RTT を測定し輻輳ウィンドウを持たせる。ある経路でタイムアウトが発生したらある一定時間その経路を使った通信を中断し、再送パケットはその他の経路を使って送る。この方法はより厳密な輻輳制御が可能であるが、輻輳制御が経路情報に依存するため SHAK E を A, B 層に分離できなくなる問題がある。

後者では SHAK E A 層で TCP の輻輳制御に相当する制御を行う。ただし、タイムアウトが発生した場合ウィンドウ幅を式 (4) にしたがって減少させる。(W_{ci}: 輻輳ウィンドウ, i: 経路識別子)

$$W_{ci} \leftarrow kW_{ci} \quad (0 < k < 1) \quad (4)$$

通信回線共有方式では複数経路を利用しているため、タイムアウトが発生しても再送パケットが宛先に送られる可能性は単一経路で通信する時よりも高い。ただし、SHAK E A 層では経路には関知しないため再送パケットが通る経路を制御できないという欠点がある。

4.2 IP の拡張

本方式を実現するために前節の SHAK E B 層と同様に、IP 層に経路情報管理テーブルを用意する。送信ホストは自分が持つ外部通信路に流すパケットはそのまま宛先ホストへ送信し、クラスタ内のホストを経由するパケットは IP 層の経路情報管理テーブルをもとにカプセル化される。

また、クラスタ内のホストは IP 層に重み付け公平キュー (WFQ) を持つ (図 4)。WFQ はホストがリレーホストとして動作するときに利用される。リレーホストが他のクラスタ内ホストからのパケットを受信すると、リレーホストの IP 層は各クラスタ内ホストに対応するキューにパケットを入れる。キューに蓄えら

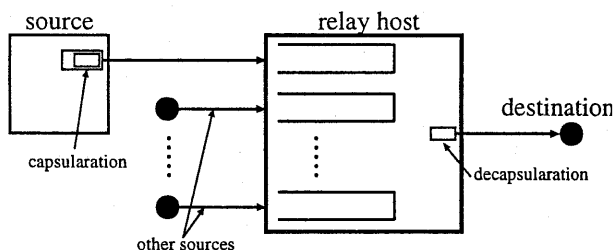


図 4 IP 層の拡張イメージ

れたパケットは各ホストに割り当てられた重みに従って順番に宛先ホストへ送信される。各キューがいっぱいになると、リレーホストはパケットの送信ホストにチャックパケットを送り送信ホストからのパケットの送信を抑制することで輻輳制御を行う。複数のホストで回線を共有する時、各通信路の帯域を各ホストに公平に割り当てるのに WFQ は有効な手法である。なぜなら、自分が持っている通信路の帯域は優先的に利用したいなどのユーザからの要求があることが予想されるからである。

本方式はある時点で与えられた複数経路を同時に利用することを目的としている。したがって、従来パケットのルーティングを担当する IP 層を、複数経路を並列利用できるように拡張できれば上位層の TCP は経路情報に関知することなく信頼性を保証することが可能である。

IP を拡張するアプローチは前節の SHAKE B 層にも適応可能である。本方式を利用した場合、クラスタ内のホストを中継して宛先ホストまでパケットを届けるためには中継ホストでのアドレス変換が必要である。したがって、どちらのアプローチをとってもパケット振り分けには IP トンネリングを用いる必要がある。

4.3 考察

TCP を拡張するアプローチは従来の TCP よりも高信頼な無線通信を可能とし、かつスループットを向上させることができると考えられる。しかしながら、信頼性保証と経路情報の管理を完全に独立化できない問題が残る。

一方、IP を拡張するアプローチは IP トンネリングを利用することで、パケットの振り分けは可能になる。しかしながら、上位層の TCP はパケットが通る経路には関知せずに信頼性を保証するため、無線環境に適応し、かつ複数経路の存在を考慮したより厳密な制御が困難である。

5 おわりに

本稿では、現在の SHAKE のプロトタイプの問題点について述べ、通信回線共有方式に適したプロトコル

について検討した。それぞれのアプローチの有効性、振り分け、再送制御などの詳細なアルゴリズムの理論的解析、検証は今後の課題である。

今後、再送制御について詳細なシミュレーションで評価を行う予定である。

参考文献

- [1] 飯田峰彦, 石原 進, 井手口哲夫, 水野忠則, “通信回線共有方式の耐障害性の検討”, 情処研報, Vol.99, No. ITS-3, pp.41-48, Oct 1999.
- [2] H.Mineno, K.Ohta, M.Aono, T.Ideguchi and T.Mizuno, “A Proposal of a Protocol for Sharing Multiple Paths in Cluster Type Network”, Proceedings of 5th Intl. Workshop on MoMuc '98.
- [3] 峰野博史, 青野正宏, 太田賢, 水野忠則, “クラスタ型ネットワークにおける通信回線共有方式の実装”, マルチメディア; 分散; 協調とモバイル (DICOMO'98) シンポジウム, Vol.98, No.8, pp.677-684.
- [4] 林 孝典, 山崎真一郎, 森田直人, 相田 仁, 武市正人, 土居範久, “TCP/IP パケットを複数経路に分配して通信する方式の性能評価”, 第 59 回情処全大, 5U-1, pp.497-500, 1999.
- [5] P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan “WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks” In *Proceedings of MOBICOM '99* pp.231-241, August 1999.
- [6] M.Mathis, J.Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgement options. Internet Request for Comments 2018, October 1996.
- [7] Vincent D. Park and M. Scott Corson. “A highly adaptive distributed routing algorithm for mobile wireless network.” In *Proceedings of INFOCOM'97*, April 1997
- [8] Josh Broch, David B. Johnson, and David A. Maltz. “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks.” Internet-Draft, draft-ietf-manet-dsr-00.txt, March 1998. (Work in progress)
- [9] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols.” In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, October 25-30, 1998, Dallas, Texas, USA.
- [10] 河口信夫, 片桐秀樹, 内柴道浩, 外山勝彦, 稲垣康善, “モバイル環境下での自律分散通信の実現とその応用”, マルチメディア; 分散; 協調とモバイル (DICOMO'98) シンポジウム, Vol.98, No.8, pp.619-626, 1998.