

複数無線リンク上での効率的コネクション型通信手法の提案

飯田峰彦[†] 橋本豊大[‡] 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。我々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。本論文では通信回線共有方式に適した複数無線リンク上での効率的コネクション型通信手法を提案する。

Proposal of the effective connection-type communication strategy on multiple wireless links

Minehiko Iida[†], Toyohiro Hashimoto[†], 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 propose the effective connection-type communication strategy on multiple wireless links.

1 はじめに

今日のノートPC、PDAといった情報携帯端末の小型化、高性能化および、携帯電話、PHSといった無線通信インフラの爆発的な普及と通信速度の向上により、出先や移動中であっても気軽に、誰とでもマルチメディアコンテンツを交換することが一般的になってきた。また、携帯電話のユーザに提供されるサービスの多様化によってモバイルコンピューティングを取り巻く環境もさらなる躍進を続けている。しかしながら、無線通信は速度が向上したとは言え、有線通信に比べると依然として帯域が狭く、品質が不安定であるという特性があるため、大量のファイル転送やリアルタイム性を要求されるようなマルチメディアデータの通信は困難である。

そこで、筆者らは携帯電話などの無線通信回線を持

つ携帯情報端末が集まり、それぞれが持つ無線通信帯域を共有することで各端末が広い帯域を使用可能にし、より信頼性のある通信を実現するための方式として通信回線共有方式[1]を提案している。

複数の経路を並列利用することで転送レートの向上を図る研究として、輻輳経路データ転送方式[3]がある。この方式は、通信を行うサーバ、クライアント間に専用のゲートウェイを設けデータはゲートウェイでIPフィルタリングによりアドレスを変換し、複数経路にパケットを振り分けるものである。これに対し、通信回線共有方式はパケットの振り分けは通信を行いたい携帯端末が行うため、専用の装置を別途用意する必要はない。また、複数経路は携帯端末の構成によって動的に変化する。

データリンク層での並列配信を実現する研究として、384KbpsのPHS接続装置を用いた通信方式[5]や、PPPマルチリンクが挙げられる。これに対し、本稿で論じる通信回線共有方式はトランスポート層での複数経路をサポートする。

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

[‡] Faculty of Information, Shizuoka University

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これまで使われてきた固定ネットワークにおける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 [6, 7] 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。これに対し、通信回線共有方式では無線通信回線を持った携帯端末の動的なメンバ構成を管理するためのプロトコルとして SCMP (*Shake Control Message Protocol*) [4] を提案している。SCMP は携帯端末間でメッセージを交換するためのプロトコルであり、携帯端末の途中離脱、途中参加を可能にする。

筆者らは、通信回線共有方式を実現するためのプロトコル SHAKE (*Shared multilinK procedures for a cluster type network Environment*) を開発している。SHAKE はプロトタイプの実装を行った結果、複数の回線を並列利用することで、転送レートが向上することを確認している。しかしながら、同じ転送速度の経路を n 本利用したとき、転送速度は n 倍にはならなかった。原因としては、パケットの振り分けアルゴリズム、複数の TCP コネクションを利用することによる受信側でのパケット再整列処理のオーバーヘッドが考えられる。

本稿では、プロトタイプの問題点、SHAKE 実現上の課題について述べ TCP を拡張するアプローチのもとでより効率的な複数無線リンク上でのコネクション型通信手法を提案する。

2 通信回線共有方式

2.1 概要

通信回線共有方式 (図 1) は、インターネットなどのネットワークへの無線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筆者らはクラスタ、クラスタを構成する携帯端末をクラスタメンバと呼ぶ。クラスタメンバ間の接続には有線 LAN や、無線 LAN, Bluetooth といった高速無線インタフェースなどの利用が予想される。各端末は 2 つのネットワーク (クラスタとインターネット) に接続するため、ネットワークインター

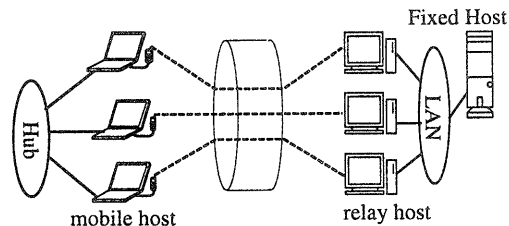


図 1 通信回線共有方式の構成例

フェースを 2 つ持つ必要がある。また、インターネット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クラスタメンバが選択可能
- 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することで伝送誤りを通信路によって独立化でき誤り訂正が容易

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。

3 プロトタイプの問題点

現時点での SHAKÉ 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 [2]。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 の間でコネクションを確立して仮想的に一つの経路の信頼性を保証するのが理想である (図 2)。しかし、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もの以外は、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 コネクションおよびクラスタと宛先ホストをつなぐ TCP コネクションの 2 つの TCP コネクションを通過する。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 コネクションは全体で $2n - 1$ 必要になる (図 3)。TCP における信頼性の保証は TCP コネクション単位で管理されるため、各コネクション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コネクション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コネクション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制御が困難である。

パケット再整理

プロトタイプのようなコネクション環境 (図 3) ではクラスタ内の端末を接続する TCP コネクション、およびクラスタと外部の間の TCP コネクションでパケットの再整理が行われている。さらに、SHAKÉ 層で複数の TCP コネクションから得られるパケットの再整理も行う必要がある。例えば、図 3 では各経路毎に 2ヶ所で再整理が行われる。ネットワーク全体としては、各 TCP コネクションのエンドで再整理が行われ、最後に

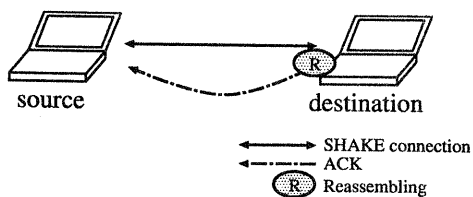


図 2 SHAKÉ コネクション

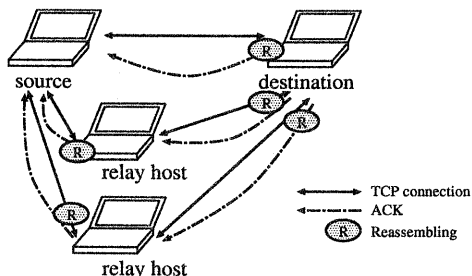


図 3 TCP コネクション (SHAKÉ プロトタイプ)

宛先の SHAKÉ プロトコルで複数経路から得られた情報からさらに再整理を行い、合計 6 回の最整理処理が必要である。

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 コネクション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 の再送処理はコネクション毎に行われているため、あるコネクション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 問題解決のアプローチ

TCP コネクションを利用することの問題を解決するためのアプローチとして

- TCP で複数コネクションを扱えるよう拡張
- IP でパケットを振り分け

が挙げられる。

前者のアプローチは複数経路の特性を考慮した再送・振り分け制御が可能である。しかしながら、通信回線共有方式を利用するためには必ず TCP を利用しなくてはならないという制約が生じる。

後者ではパケットを振り分けるために IP トンネリングを使用する必要がある。パケットの振り分けを経路制御を行う IP 層で行うため、上位層は振り分けに関知せずに制御を行うことができる。また、TCP, UDP のどちらでも通信回線共有方式を利用できる。

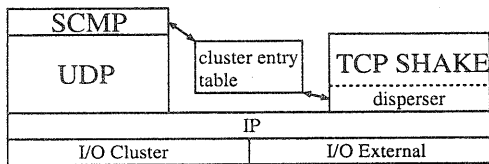


図 4 TCP SHAKE プロトコルスタック

本稿では、前者の TCP を拡張するというアプローチのもとで効率的な通信回線共有方式の実現手法を提案する。

5 TCP SHAKE

通常、TCP では 1 つの送信バッファとそれに付随する送信ウィンドウ、輻輳ウィンドウ、RTT 計測用の変数、再送タイマによりデータの送信を管理している。一方、TCP SHAKE では複数の経路を効率的に利用して輻輳制御、再送制御を行うため、各経路毎に

- 輻輳ウィンドウ
- RTT
- 再送タイマ

を用意する。各経路毎にこれらの値を管理するため、TCP SHAKE では ACK の返送に受信側が受信したセグメントと同一の経路を用いる。このため、Delayed ACK は使用しない。RTT、再送タイマは経路ごとに計測、設定する。smoothed RTT、再送タイムアウトの算出アルゴリズムは従来の TCP と同様の方法を採用する。また、セグメントの振り分けに SCMP によって生成されるクラスタエントリテーブルの情報を利用する。セグメント送信時のウィンドウ幅は各経路の輻輳ウィンドウの和と受信側の受信バッファの小さい方を送信ウィンドウとする。

5.1 振り分け処理

TCP では、セグメントと呼ばれる単位でデータを送信する。TCP の最大セグメントサイズは 1460(byte) である。各セグメントには TCP のヘッダが付加され、IP 層に渡される。SHAKE TCP では、クラスタエントリテーブルをもとに複数存在する経路にセグメントを振り分ける。クラスタを構成する各ホストが保有している無線通信路の帯域幅、エラーレートといった通信特性はそれぞれ異なる。このため、セグメントの振り分け方式がスループットに影響する。振り分け方式としては

- 各経路に等分配
- 帯域幅を重みにして各経路に振り分けるセグメントを制御

の 2 つの方式が考えられる。前者はアルゴリズムが簡単であり、実現しやすいという利点はあるが、スループットが最も帯域の狭い経路に依存してしまう上に受信側で適切な順序で受信できないため不必要な再送が頻繁に発生し、性能を悪化させる可能性がある。後者はアルゴリズムは複雑になるが、受信側で適切な順序で受信することができ、性能の向上を期待できる。

そこで、クラスタエントリテーブルから各経路の通信帯域幅の情報を取得し、これを振り分け時の重みとして利用する (図 4)。経路の重みにしたがって輻輳ウィンドウの初期値を決定し、各経路の輻輳ウィンドウ幅でセグメントを振り分ける。輻輳ウィンドウについては次節で詳しく述べる。

5.2 輻輳制御

輻輳制御には送信ウィンドウと各経路毎の輻輳ウィンドウを利用する。また、輻輳ウィンドウはセグメントが正しく送信されるたびに (ACK が 1 つ返送されるたびに) 変動するスロースタートを利用する。経路が確立されると各輻輳ウィンドウを初期化する。このとき、輻輳ウィンドウ w_i はクラスタが保有する無線通信路の帯域幅の構成によって変化する。 w_i は以下のように決める。まず、帯域幅が最小の経路のウィンドウ幅を 1 セグメント長 (MSS) とする。それ以外の経路のウィンドウ幅は以下の式で算出する。

$$w_i = \left\lfloor \frac{B_i}{B_{min}} \right\rfloor \cdot w_j \quad (i \neq j) \quad (1)$$

($\lfloor x \rfloor$ は x を越えない最大の整数、 j は帯域が最小の経路)

上式は、輻輳ウィンドウの比が通信帯域幅の比にほぼ一致していることを示している。初期ウィンドウ幅が設定された後、各経路の輻輳ウィンドウは ACK が返ってくる度にウィンドウ幅を初期ウィンドウ幅分だけ増加する。ただし、従来の TCP のスロースタートと同様、指数的に増加する範囲と線形に増加する範囲の閾値 (ssthresh) を設ける。ウィンドウ幅が閾値を越えた後はウィンドウ幅の増分は初期ウィンドウ幅よりも小さくなる。ウィンドウ幅の変動を式で表すと以下のようになる。

$$w_{i+1} = \begin{cases} w_i + w_0 & (w_i < ssthresh) \\ w_i + w_i * w_i / MSS & (w_i > ssthresh) \end{cases} \quad (2)$$

セグメント送信中にある経路でタイムアウトが発生すると、その経路のウィンドウ幅は初期値に減少する。なお、 w_i の最大値は

$$\sum w_i < snd_wnd \quad (3)$$

に従うものとする (snd_wnd : 送信ウィンドウ)。TCP SHAKE ではセグメントはそれぞれ異なる特性を持った経路を通るため、送信したセグメントの順序と受信側で受信する順序は一致しない状況が多々起こり得る(顕著に現れる)。受信側で正しい順序でセグメントが受信されないと重複 ACK が頻繁に発生し、無駄な再送をひきおこし、その結果スループットは著しく低下する。このため、性能は送信側でのセグメント振り分け方式に大きく左右される。

振り分けは帯域が広い経路には多くシーケンス番号が小さいセグメントを、狭い経路には少なくシーケンス番号が大きいセグメントを振り分けるのが望ましい。輻輳ウィンドウ幅の初期値を従来の TCP のように一律最大セグメント長にするのではなく、異なる特性を持つ複数経路の構成によって変動させるようにしたのは、通信開始時から先に述べたような振り分けができるようにするためである。

5.3 再送制御

再送タイムアウトは、各経路で測定している RTT をもとに決定する。ある経路でタイムアウトが発生すると、その経路の輻輳ウィンドウは初期値に設定される。セグメントを再送する時、再送セグメントは振り分け方式に従い、帯域が最大の経路で送信する。そして、再送するセグメント数は帯域が最大の経路のウィンドウ幅に従う。従来の TCP ではタイムアウトが発生するとタイムアウトは $2n$ 倍になる (n は連続してタイムアウトになった回数)。経路が単一の場合は輻輳が発生していると判断し、再送セグメントはより確実に送るために長いタイムアウト時間待つのは妥当であるが、TCP SHAKE では各セグメントは異なる経路を通して受信側に送られるため、セグメントが不連続に受信される可能性が単一経路での通信よりも高くなる。セグメントを断続的に受信すると、重複 ACK が頻繁に発生しスループットが低下する。そこで、TCP SHAKE では SACK(Selective Acknowledgement) を利用する。SACK はセグメントの受信側が断片的に受信したセグメントブロックの情報を ACK の TCP オプションフィールドに付加することで、送信側により詳細な受信状況を知らせることによって、タイムアウト発生時の無駄なセグメント再送を防ぐものである。

表 1 シミュレーション条件

| | 再送方式 | 輻輳ウィンドウ | SACK |
|------|------|---------|------|
| 方式 A | ○ | ○ | × |
| 方式 B | × | ○ | × |
| 方式 C | ○ | ○ | × |
| 方式 D | ○ | × | × |
| 方式 E | × | ○ | ○ |
| 方式 F | × | ○ | × |

再送方式 ○: 帯域が最大の経路で再送
 ×: タイムアウトが発生した経路
 輻輳ウィンドウ ○: 帯域に応じた増分
 ×: 増分は同一
 SACK ○: 利用する
 ×: 利用しない

SACK を利用し、複数経路を効率的に利用する、すなわち適宜セグメントを送信する経路を切替えることで同じセグメントの連続再送は極力防ぐことができ、単一経路の場合に比べて性能の向上を期待できる。

6 性能評価

6.1 シミュレーション

提案方式に関する性能評価をシミュレーションより行った。シミュレーションでは、TCP SHAKE が 3 つの経路で通信を行うことを想定した。各経路の通信帯域幅はそれぞれ、64Kbps, 64Kbps, 32Kbps である。受信側の受信バッファは 36500(byte)、パケットはランダムでロスさせた。この条件下で表 1 に示す方式で 100 から 1000(Kbyte) のデータを送信したときのそれぞれの送信時間を測定した。

6.2 結果と考察

シミュレーション結果を図 5, 図 6, 図 7 に示す。図 5 は方式 A と方式 B, 図 6 は方式 C と方式 D, 図 7 は方式 E と方式 F の結果を示している。

再送方式に関しては(図 5), 再送セグメントを送信する経路を帯域が広い経路に切り替えることによる性能向上が見られた。パケットの振り分けや、再送時に利用する、無線通信路の特性を示すパラメータは帯域とパケットロス率が挙げられる。今回は再送セグメントを送信する経路の判断基準として、コネクション確立時にクラスタエントリテーブルから得られる帯域幅

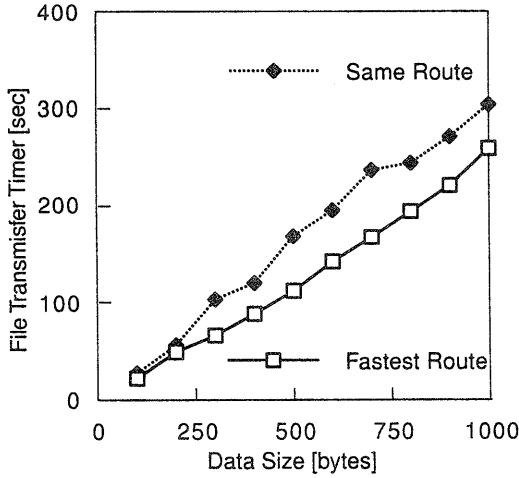


図 5 再送経路の選択の効果

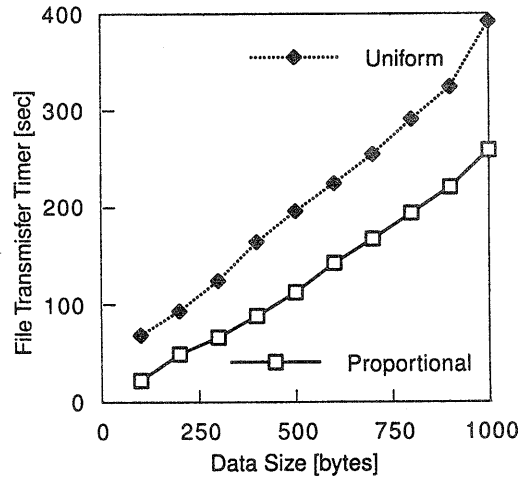


図 6 経路毎の制御手法の効果

の情報のみを利用したが、帯域は広いがパケットロス率が高いような通信環境ではこの再送方式は適用できない。したがって、再送や振り分けの判断基準となりうる経路の重みは帯域だけではなく、パケットロス率も考慮する必要がある。重み算出のための具体的なアルゴリズムは今後の検討課題である。

輻輳ウィンドウ更新時の増分に関しては(図 6)、各経路の帯域に応じて輻輳ウィンドウの増分を変化させることで転送速度が向上した。しかしながら、TCP で標準的に使われている受信バッファサイズで利用すると、各経路の輻輳ウィンドウ幅の和の方が受信バッファよりも大きくなってしまい、送信時のウィンドウ幅は受信バッファサイズで固定になってしまうため、動的な輻輳ウィンドウ制御が働かなくなる。したがって、TCP で標準で使われているよりも TCP SHAK には受信バッファサイズを大きくする必要がある。

SACK の利用に関しては(図 7)、顕著な差が見られなかった。SACK は受信側が断片的にセグメントを受信することによる重複 ACK、重複再送を回避する効果を与えるが、それは受信側のセグメント受信状況、パケットロス率に依存すると考えられる。

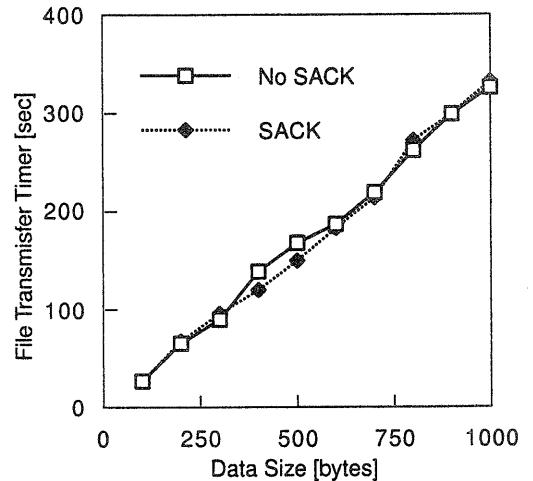


図 7 SACK 使用の効果

7 おわりに

本論文では、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効率的なコネクション型通信手法について提案し、提案手法の性能評価を行った。その結果、輻輳制御、再送制御に複数経路を効率的に利用することで性能が向上することを確認した。今後、さらに詳細な評価を行っていく。

参考文献

- [1] 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.
- [2] 峰野博史, 青野正宏, 太田賢, 水野忠則, “クラスター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.
- [3] 星谷直哉, 相田 仁, 斎藤忠夫, “複数経路データ転送におけるコネクション指向プロトコルの提案,” 第 59 回情処全大, 1T-7, pp.349-350, 1999.
- [4] 橋本豊大, 飯田峰彦, 石原 進, 水野忠則, “SCMP: 通信回線共有方式 SHAKE における制御メッセージプロトコル”, マルチメディア; 分散; 強調とモバイル (DICOMO 2000) シンポジウム, Vol.2000, No.7, pp.385-390.
- [5] 神尾享秀, 児島史秀, 藤瀬雅行, “384kbps-PHS 実験装置の概要と性能評価,” 情処研報 Vol.99, No.80, 99-MBL-10, 1999.
- [6] Vincent D. Park and M. Scott Corson. “A highly adaptive distributed routing algorithm for mobile wireless network.” In *Proceedings of INFOCOM'97*, April 1997
- [7] 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)