推薦論文

通信回線共有方式を用いた高速 Web アクセスの実現

小西洋祐[†] 石原進[††] 水野忠則[†††]

現在いつでもどこでもだれでもインターネットに接続することができるモバイルコンピューティング環境が急速に普及し，より身近なものとなってきた．しかしながら無線通信環境は有線のそれに比べると通信の品質が低く，通信速度が遅いという問題がある．そこで筆者らは数台の移動ホストにより一時的に短距離のネットワークを構築し，これらの各端末が持つ長距離無線リンクを束ねることで論理的な帯域幅を広くし，より品質の高い通信環境を実現するための方法として，通信回線共有方式 SHAKE を提案している．SHAKE はこれまでに試作システムが実装・評価されているが，SHAKE を利用するすべての端末に機能を追加する必要があった．本稿では通信回線共有方式を利用した高速 Web アクセス方式として Web SHAKE を提案する．Web SHAKE は各移動ホスト上で動作する HTTP Proxy サーバを利用することにより，SHAKE のための特別な機能を有しないインターネット上の任意の Web サーバと SHAKE による通信が可能である．本稿では Web SHAKE の実装・評価結果について示す．

Fast WWW Access with Sharing Multiple Paths Procedure for Cluster Network Environment

Yousuke Konishi,[†] Susumu Ishihara[††] and Tadanori Mizuno[†††]

Nowadays mobile computing has become popular. Many people can access to Internet with mobile terminals. However wireless links used by mobile hosts have some problems such as narrow bandwidth and low reliability. To offer high speed communication on wireless links, we have proposed *SHAKE* (SHAring multiple paths procedure for cluster networK Environment). In SHAKE, mobile hosts which are connected with fast local link each other use multiple wireless links owned by each host simultaneously to communicate with hosts on the Internet. An experimental system has been implemented and evaluated. In this system, however, not only mobile hosts but also correspondent hosts on the internet require special software for SHAKE. In this paper, we propose a fast WWW access method with SHAKE (*Web SHAKE*). The feature of Web SHAKE is that it does not require any special software on web servers on the internet to use SHAKE. The feature is realized by an use of HTTP Proxy Server which works on each mobile host. Experimental results obtained by an implementation of Web SHAKE are shown.

1. はじめに

パソコンの普及，通信環境の発展によりインターネットに接続するユーザが爆発的に増加している．またノートパソコン，PDA などの小型携帯端末，PHS や携帯電話の普及によりモバイル環境も発展しており，モバイル環境からインターネットに接続するユーザも現在では珍しくない．

そのような中でインターネット接続者の増加にともないインターネットに求められるニーズも多様化し，より高品質なサービスが求められている．WWW（World Wide Web）を例にとると，画像や音声，動画などのマルチメディアデータを使ったホームページは現在では珍しくなく，コンテンツの大容量化が進んでいる．このようにデータが肥大化する一方，無線通信は有線通信に比べると帯域幅が狭く，マルチメディアデータなどの大容量のデータ転送にはいまだ十分と

† 静岡大学大学院情報学研究科
Graduate School of Information, Shizuoka University
†† 静岡大学工学部システム工学科
Faculty of Engineering, Shizuoka University
††† 静岡大学情報学部
Faculty of Information, Shizuoka University

本稿の内容は 2001 年 2 月の第 16 回 MBL 研究会にて報告され，MBL 研究会運営委員により情報処理学会論文誌への掲載が推薦された論文である．

はいい難い．また無線環境ではハンドオフやビットエラーにともなう TCP の性能の悪化などにより，さらにデータのダウンロードの時間が増すことになる．

筆者らはこれまでに無線通信環境下でも高速，高品質のデータ通信を実現するための方式として通信回線共有方式 SHAKE（SHaring multiple paths procedure for cluster networK Environment）を提案している．これは移動端末それぞれが持つ無線リンクを共有し，一時的にネットワークを構成してそれらを論理的に束ねることにより，高速・高品質なデータ通信を実現する方式である．筆者らはこれまでに SHAKE の試作システムを実装，性能評価を行っている[1]．しかしながら試作システムにおける SHAKE の機能は TCP 上で動作するアプリケーションプログラムで利用するライブラリとして実現されている．この実装法では無線リンクを共有する移動端末のみならず，移動端末による一時的ネットワーク外のホスト上のアプリケーションにもこのライブラリを組み込む必要がある．このため，このライブラリを組み込んだアプリケーションを持たないホストとの無線リンクの共有ができないし，外部ホストとは SHAKE による通信ができないという問題がある．

本稿では利用するサービスを WWW に特化することにより，インターネット上の任意のホストと SHAKE の利用を可能とする実現法 Web SHAKE を提案する．Web SHAKE では，各移動端末上で動作する HTTP Proxy サーバを利用することにより，通信回線を共有する移動端末へのプログラム追加のみで，特別なソフトウェアを備えないインターネット上の任意のホストへの高速な Web アクセスを実現する．本稿ではこの WebSHAKE を提案し，その実装・評価について述べる．

2. 通信回線共有方式

2.1 概　　要

現在外出先や移動中にインターネットに接続する場合，PHS，携帯電話を使用したデータ通信サービスが利用されるが，これらは有線通信に比べると通信速度も遅く，品質も悪い．そこでそれぞれ外部への無線リンクを持つ複数の移動端末が一時的にネットワークを作り（このネットワークをクラスタと呼ぶ），ある端末がクラスタ外のホストと通信するときは，それ自身が持つ無線リンク以外にクラスタ内の他の移動端末の持つ無線リンクも利用すれば，単体では低速・低品質な無線リンクしか持たない端末でも高速・高品質な通信が可能である．この方式を筆者らは通信回線共有方式（SHAKE）と呼んでいる．SHAKE による通信の例を**図 1** に示す．SHAKE には以下のような利点がある．

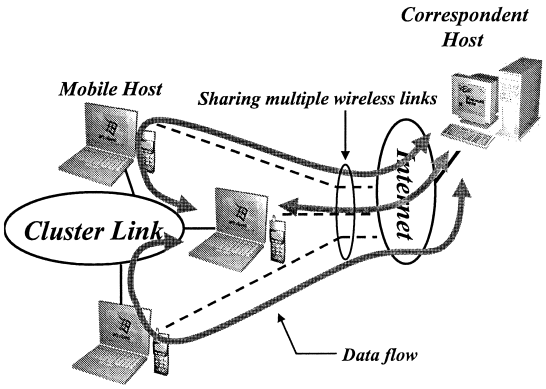

図 1 通信回線共有方式（SHAKE）を利用した通信例
Fig. 1 Example of communication in using SHAKE.

- 無線通信環境下での転送レートの向上．
- 自身が外部へのリンクを持たないとき，または利用不可能なときでも他の端末のリンクを利用可能．
- クラスタは動的に形成することができ，物理的な場所に依存しない．

関連研究として PPP マルチリンク[2] や専用装置を使用した複数 PHS の同時利用方式[3] などがある．これらはいずれも個々のホストが持つ複数のネットワークインタフェースを同時に利用してデータリンク層で並列化を行うものであり，SHAKE はネットワーク層以上での並列化を目指す点でこれらとは異なる．

SHAKE は複数の携帯端末の持つ無線通信路を一時的に共有して通信を行うという概念を示すものであり，その実現には，TCP，IP 層での実現や，複数の TCP コネクションを利用したアプリケーション層での実現などが考えられる．これまでに SHAKE を実現した例の 1 つとして SHAKE 試作システムがこれまでに実装・評価されている[1]．これは複数の TCP 接続を利用したアプリケーションとしての実現である．SHAKE の機能は TCP を用いたライブラリとして実装されており，複数経路へのトラフィック分配，分割されて到着したデータの再整列・結合処理などを行っている．

2.2 試作システムの問題点

SHAKE を実現するためにはクラスタを構成する移動端末と相手先のクラスタ外のホストとの連係を必要とする．たとえば移動端末がある相手ホストからあるファイルを複数のリンクを同時に利用して受信する場合，相手ホストはデータを分割し，それぞれ別の経路にデータを振り分けなければならない．このため相

手ホストはクラスタを構成する各移動端末の情報を管理する必要がある．またクラスタ内の移動端末側では，分割されたデータをファイル転送要求を出した移動端末に集め，それらを再整列・結合しなければならない．SHAKE 試作システムで用いた実現方法では，SHAKE を実現するためにクラスタを構成する各移動端末，そして通信する相手先ホストなど通信するすべてのホストに SHAKE を実現するモジュールを実装していた．このため SHAKE の機能を組み込んでいないホストと通信できないという問題がある．

3. Web SHAKE

3.1 概　　要

試作システムで用いた実現方法では，クラスタと通信するクラスタ外部のホストは，SHAKE 用のソフトウェアを組み込んだものに限定されていた．しかしながら SHAKE を利用するうえでは，任意のホストと通信ができる方が望ましい．そこで利用するプロトコルを HTTP（Hyper Text Transfer Protocol）に限定することで，インターネット上の任意のホストと SHAKE による通信が可能な高速 Web アクセス方式 WebSHAKE を提案する．Web SHAKE では HTTP Proxy サーバを利用することによりクラスタを構成する各移動端末のみに SHAKE を実現するモジュールを組み込むだけで，任意の Web サーバと通信することが可能である．

3.2 WebSHAKE アーキテクチャ

HTTP [4),5)] にはファイルを受信するリクエストメッセージである GET 要求以外に POST 要求など種々のメッセージがある．WebSHAKE では受信するファイルのサイズ情報，更新情報などを含んだ HEAD 情報のみを受信する HEAD 要求と HTTP1.1 から取り入れられた受信するファイルの範囲を指定できる領域指定要求（Partial GET）を利用する．HEAD 要求は受信するファイルのサイズ情報を取得するために，Partial GET はファイルを分割して受信するために使用する．

WebSHAKE のアーキテクチャを**図 2** に示す．クラスタ内の各移動端末で SHAKE の機能を組み込んだ SHAKE HTTP Proxy（以下 SHP）を動作させる．SHP がローカルで動作する HTTP クライアントから HTTP 要求を受信し，他の移動端末上で動作する SHP へ HTTP 要求を分散させる．ローカルホスト以外から要求を受信した場合，SHP は通常の HTTP Proxy として振る舞う．すなわちクラスタ内の他の端末から分散された要求を受信した SHP は，受信した

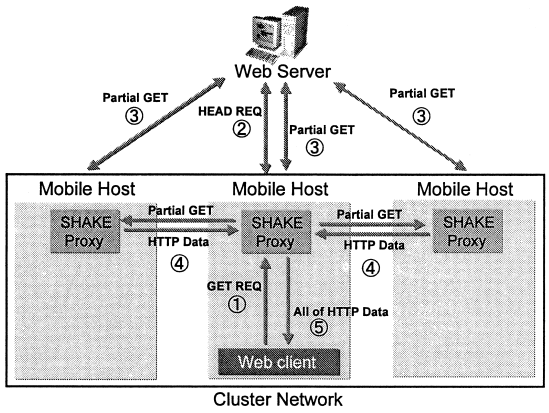

図 2 WebSHAKE アーキテクチャ
Fig. 2 Web SHAKE architecture.

HTTP 要求をそのまま Web サーバへ代理要求することになる．以下 WebSHAKE における Web サーバからクライアントへのデータ転送プロセスを説明する．

(1) SHP はローカルホストで動作する Web ブラウザなどの HTTP クライアントから GET 要求を受信する．
(2) SHP は受信するファイルのサイズ情報を取得するために，Web サーバへ HEAD 要求を送信してファイルサイズを取得する．
(3) SHP は，取得したファイルサイズをもとに，各無線リンクに割り当てる転送ファイルサイズを決定し，Partial GET 要求をクラスタ内の各移動端末上の SHP に送信する．各 SHP は並列に動作し，Web サーバからのファイル取得を Partial GET 要求により行う．
(4) SHP は HTTP Proxy サーバとして振る舞うので，受信した分割ファイルはそのまま GET 要求を発したクライアントが動作する移動端末の SHP へ送信される．
(5) クライアントから GET 要求を受けた SHP は，分割して受信したファイルのデータを再整列・結合し，HTTP クライアントへ送信する．このとき分割して受信したファイルには不要なヘッダが付加されているのでこのヘッダを取り除き，(2) で取得したオリジナルヘッダを付加してから結合する．

WebSHAKE では SHP が Partial GET 要求を利用することで，クラスタ側で Web サーバからのデータ受信経路を振り分けることができるため，Web サーバがクラスタ内の移動端末を管理したり，複数経路へのデータ分配を行ったりする必要はない．また SHP が HTTP Proxy サーバとして振る舞うので，クラス

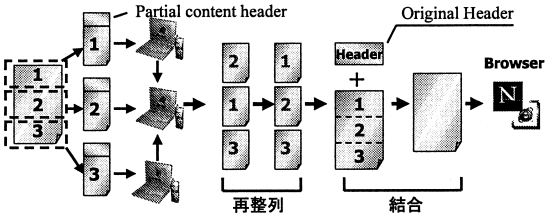

図 3 WebSHAKE におけるファイルの再整列・結合
Fig. 3 Re-queuing and combine download file in Web SHAKE system.

タ内の移動端末でも SHAKE 用の特別な HTTP クライアントなどを用意する必要がない．

3.3 SHAKE HTTP Proxy の役割

3.3.1 複数経路の管理

Web SHAKE では 1 つのファイルを分割して同時に複数のリンクを介して受信するので，クラスタ内の移動ホストは，接続先の Web サーバ以外に，クラスタを構成する各ホストとも通信する必要がある．SHP は接続先の経路やクラスタ内の各ホストへの経路を**クラスタテーブル**で一括して管理する．クラスタテーブルの各レコードにはホスト名，ポート番号，振り分け比率のフィールドがある．各経路からは同時にデータが流れてくるので，SHP はつねに流れるデータの in/out を監視し並列に受信するよう設計されるべきである．

3.3.2 ファイルの再整列・結合処理

Web SHAKE では 1 つのファイルを複数のデータに分割して受信する．このため SHP は受信した分割ファイルを 1 つに結合する必要がある．**図 3** にその様子を示す．Partial GET 要求によって分割して受信したデータの先頭には PartialContent という不要なヘッダが付加されている．このため SHP はまず各分割ファイルの不要なヘッダを取り除いた後にこれらを結合し，HEAD 要求で受信したオリジナルヘッダを付加する．

また各経路の帯域幅や遅延は各回線により異なるので，必ずしもファイルの先頭から順に分割ファイルが受信されるとは限らない．たとえば 1000 [bytes] のファイルの要求後，0-500 [bytes] を HOST1，501-999 [bytes] を Host2 経由で受信したとき，Host2 の通信速度が速いときは，501-999 [bytes] が最初に受信されるときもある．したがって SHP はこれらの分割ファイルが誤った順序で受信されても，正しい順序に整列する．

4. 性 能 評 価

4.1 性能の理論的解析

WebSHAKE の性能を理論的に解析する．前提としてすべての移動端末（MH: Mobile Host）と Web サーバ間の遅延および帯域幅はそれぞれ等しいとする．またすべての MH で SHP が動作するものとする．ここでは TCP 接続確立時のオーバヘッドは無視している．この前提に基づき，記号を以下のように定義する．

n	クラスタ内の MH の数
d_{w}	MH と Web サーバ間の遅延
d_{c}	クラスタ内の MH 間の遅延
d_{h}	ローカルホスト内の Web クライアントと SHAKE Web Proxy 間の通信遅延
B_{w}	MH と Web サーバ間の帯域幅
B_{c}	クラスタ内の帯域幅
B_{h}	ローカルホスト内の Web クライアントと SHP 間の通信帯域幅
H	HTTP ヘッダサイズおよび HTTP 要求サイズ（両者等しいとする）
D	転送データサイズ
τ_{p}	SHP での処理時間
τ_{s}	Web サーバ応答時間

4.1.1 通常の WWW アクセス応答時間

通常の Web アクセスにおける応答時間 T_{n} は以下の式で表すことができる．

$$T_{\mathrm{n}} = 2d_{\mathrm{w}} + \frac{2H + D}{B_{\mathrm{w}}} + \tau_{\mathrm{s}} \tag{1}$$

この式は 1 回の往復遅延と 1 回のリクエスト転送時間，1 回のデータとヘッダの転送時間およびサーバ処理時間の和を表している．

4.1.2 WebSHAKE での WWW アクセス応答時間

WebSHAKE での WWW アクセスにおける応答時間 T_{s} は以下の式で表すことができる．

$$\begin{aligned} T_{\mathrm{s}} = {} & (5\tau_{\mathrm{p}} + 2\tau_{\mathrm{s}}) + (2d_{\mathrm{h}} + 4d_{\mathrm{w}} + 2d_{\mathrm{c}}) \\ & + (2H + D)/B_{\mathrm{h}} + (4H + D/n)/B_{\mathrm{w}} \\ & + (2H + D/n)/B_{\mathrm{c}} \end{aligned} \tag{2}$$

SHP は Web クライアントと同一のホストで動作させることを前提としているため，d_{h} は無視できるほど小さく，B_{h} は十分に大きい．このため式 (2) は以下のように書き換えることができる．

$$\begin{aligned} T_{\mathrm{s}} = {} & (4d_{\mathrm{w}} + 2d_{\mathrm{c}}) + (2\tau_{\mathrm{s}} + 5\tau_{\mathrm{p}}) \\ & + \frac{4H + D/n}{B_{\mathrm{w}}} + \frac{2H + D/n}{B_{\mathrm{c}}} \end{aligned} \tag{3}$$

簡略化のため詳細なステップは省くが，通常の

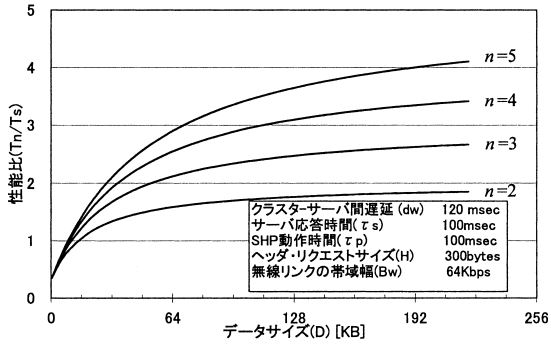

図 4 理論的な WebSHAKE の性能比
Fig. 4 Throughput predict of Web SHAKE system.

WWW アクセスとの主な違いは以下の 3 点である．

(1) SHP は Web サーバにファイルサイズ取得のため HEAD 要求を直接送り受信するファイルのヘッダを受け取る．このときのオーバヘッドが $\tau_{\mathrm{p}} + 2d_{\mathrm{w}} + 2H/B_{\mathrm{w}}$ となる．

(2) Web サーバから 1 つのファイルをクラスタ内の各 MHn 台が受信データ D を並列に受信するのに $(2H + D/n)/B_{\mathrm{w}}$ を要する．

(3) $n-1$ 台の MH が D/n で分割したデータを Web クライアントから GET 要求を受信した SHP へ転送するのに $(2H + D/n)/B_{\mathrm{c}}$ を要する．

以上のほか，各 MH における SHP の処理時間（τ_{p}），Web サーバでの処理時間（τ_{s}），そして各端末間における遅延（$d_{\mathrm{w}}, d_{\mathrm{c}}$）の総和が式 (3) である．上記でステップ (2) 以外はすべて SHAKE 特有のオーバヘッドとなる．特にステップ (3) は $B_{\mathrm{w}} \geq B_{\mathrm{c}}$ のような場合，すなわちクラスタ内の帯域幅が，MH と Web サーバ内の帯域よりも小さい場合大きなオーバヘッドとなる．

4.1.3 WebSHAKE と通常の Web アクセスとの比較

応答時間の比 $T_{\mathrm{n}}/T_{\mathrm{s}}$ を計算すると以下のような式となる．ただし SHAKE ではクラスタネットワークは共有する外部への無線リンクに比べて十分に高速であることを前提としているため $B_{\mathrm{w}} \ll B_{\mathrm{c}}$，$d_{\mathrm{c}} \approx 0$ とし，式 (3) における $d_{\mathrm{c}} = 0$ および $(2H + D/n)/B_{\mathrm{c}} = 0$ と見なす．

$$T_{\mathrm{n}}/T_{\mathrm{s}} = \frac{n + 2nH/D + n(2d_{\mathrm{w}} + \tau_{\mathrm{s}})B_{\mathrm{w}}/D}{1 + 4nH/D + n(4d_{\mathrm{w}} + 2\tau_{\mathrm{s}} + 5\tau_{\mathrm{p}})B_{\mathrm{w}}/D} \quad (4)$$

図 4 に理論的な WebSHAKE の性能比を示す．この図は式 (4) より得られる $T_{\mathrm{n}}/T_{\mathrm{s}}$ のデータサイズ D に対する変化を示している（各パラメータについては図 4 を参照）．図 4 より，サイズの小さなデータを複数リンクで受信するとオーバヘッドの方が大きくなり性能が悪化するが，サイズが大きくなるにつれてスループットが向上することが予測できる．

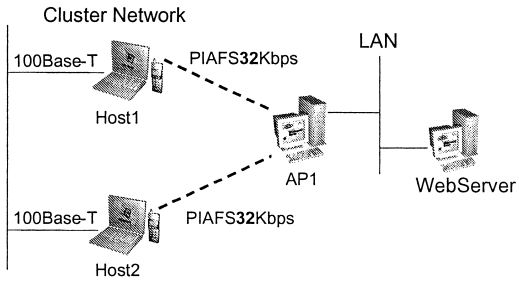

図 5 実験 1 の環境
Fig. 5 Experimental environment-1.

4.2 実　　装

WebSHAKE を実現するものとして SHP を実装した．開発言語には C を使用しており，FreeBSD，Linux などの UNIX 系 OS 上で動作する．SHP は他のホストからは一般の HTTP Proxy サーバとして振る舞うので，特別な Web クライアントは必要ない．SHP にはクラスタの数，クラスタを構成する各ホストの名前（IP アドレス），ポート番号，そして振り分けの比率を与える．現在の実装状況では各リンクの状態に応じて動的に振り分けの比率を変更させることはできない．

4.3 実　　験

4.3.1 実験環境

実験には PIAFS 通信用の PCMCIA カードを装着したノート PC3 台を使用した．ノート PC には Sharp の PJ2-X3（CPU: PentiumII 300 Mhz, Memory: 128 Mbytes）1 台と Panasonic の CF-M2（CPU: Cereron 500 Mhz，Memory: 128 Mbytes）2 台を使用し，OS にはすべて Linux Vine2.0R を使用した．PHS 通信用の PCMCIA カードには SII の MC-P210 を 2 枚と NTTDoCoMo の P-in Comp@ct1 枚を使用した．クラスタは 100BASE-T のイーサネットで構成した．

4.3.2 実　　験

以上のような環境で以下の 3 つの実験を行った．

実験 1：各リンクにばらつきがない場合の検証

無線リンクのスループットにはばらつきがなく，ほぼ同じスループットが得られる環境で性能を測定した．実験 1 の実験環境の構成図を**図 5** に示す．この環境で無線リンクを 1 本使用したときと，2 本共有したときで，Web サーバから 1 Kbytes〜1 Mbytes までの様々なサイズのファイルを受信したときの受信時間を測定した．ここで受信時間とは HTTP クライアントから

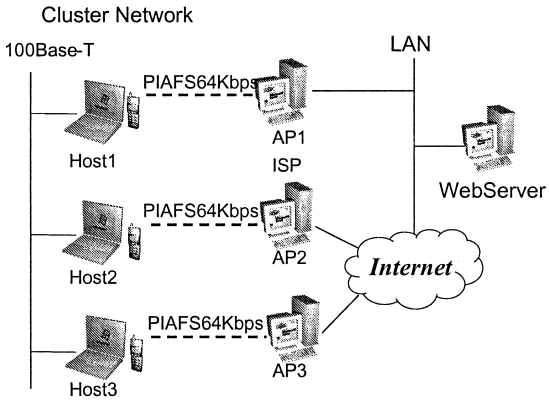

図 6 実験 2 の環境
Fig. 6 Experimental environment-2.

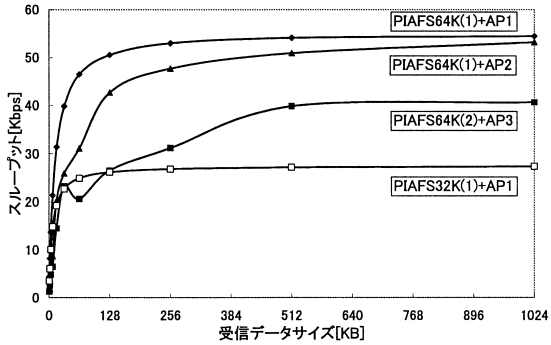

図 7 実験に使用した通信経路の性能
Fig. 7 Wireless link performance using of experiment.

の GET 要求送信から，結合したファイル全体の受信までに要した時間である．測定は各ファイルに対して 50 回測定を行い，その平均を求めた．図 5 のとおり，Access Point（以下 AP）と Web サーバには静岡大学の LAN 上のものを使用し，インターネットを経由していない．各無線リンクの通信速度は 32Kbps である．

実験 2：各リンクにばらつきがある場合の検証

AP には商用の ISP を利用しインターネットを経由させ，使用する無線リンクのスループットにばらつきがある環境で測定した．実験 2 の実験環境の構成図を**図 6** に示す．実験 2 では 64 Kbps の帯域幅を持つ無線リンクを 1〜3 本共有したときの Web SHAKE の性能を測定した．測定方法は実験 1 と同様，様々なファイルを受信したときの受信時間をそれぞれ各 50 回測定した．**図 7** に実験に使用した各無線リンクの PIAFS 通信カードと AP の組合せにおけるスループットを示す．これは静岡大学内の LAN 上の Web サーバから 1 Kbytes〜1 Mbytes のデータを HTTP で受信したときのスループットを表したものである．各無線リンクに対してトラフィックは各経路に等分配した．共有するリンクが 2 本のときに使用した無線リンクは図 7 中の PIAFS64K(1)+AP1 と PIAFS64K(1)+AP2 の 2 種類である．測定に使用した無線リンクが 1 本のときは PIAFS64K(1)+AP1 のリンクである．接続先は学内の Web サーバであり，図 6 の Host2 がデータの受信者である．

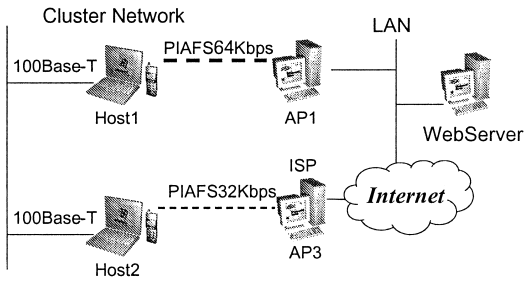

図 8 実験 3 の実験環境
Fig. 8 Experimental environment-3.

実験 3：振り分け比率の影響の検証

実験 3 の実験環境の構成図を**図 8** に示す．実験 3 では 2 本のリンクを共有して実験 1 と同様，様々なサイズのデータを受信しその受信時間を測定したが，2 本のリンクはそれぞれ最大帯域幅が異なる．1 つは 32 Kbps であり，もう 1 つは 64 Kbps である．この実験では，データを各リンクに同じ量のデータを振り分けた場合と，振り分け比率を 1 : 2 として，64 Kbps の回線には 32 Kbps の回線よりも倍の量のデータを受信するようにした場合の 2 つの条件において Web サーバからファイルを受信するまでの時間をそれぞれ 50 回測定した．

4.4 結果と評価

4.4.1 実験 1 の結果と評価

実験 1 の結果を**図 9** に示す．この図より，受信するファイルのサイズが大きいほどスループットが向上することが確かめられる．また 1 Mbytes のファイルを 2 本の無線リンクで受信したときは，1 本で受信したときと比べて約 1.85 倍の性能向上が得られている．実験 1 では図 4 で示した環境に近い状況で実験を行ったため，性能も予測したとおりの結果が得られた．

理論的な予測どおり，大きなファイルを受信するときには性能の向上がみられる反面，小さなファイルを受信しているときには性能の低下がみられる．図 9 から 16 Kbytes 以下のデータを受信したときには性能が無線リンク 1 本を使用したときよりも後退していることが分かる．この理由としては 4.1 節で検討したオーバヘッドのほかに，4.1.2 項では無視していた TCP 接続確立にともなうオーバヘッドが考えられる．

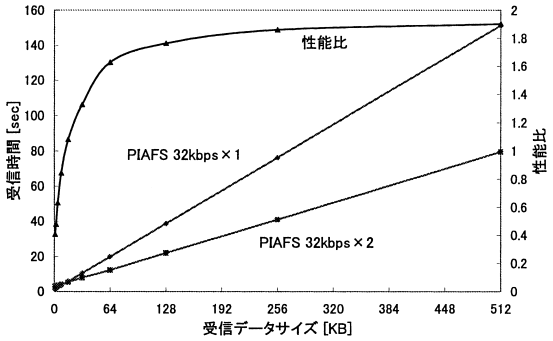

図 9 学内の AP を用いた PIAFS32Kbps × 2 の実験結果
Fig. 9 Result of experiment-1.

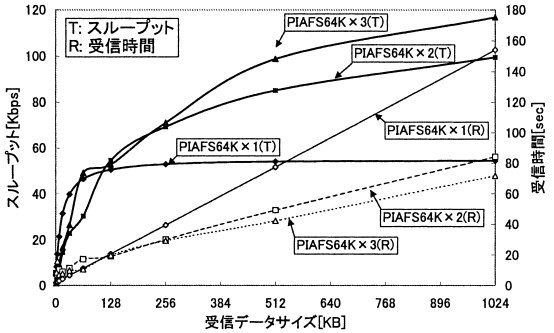

図 10 通信回線を 1〜3 本共有しての実験結果
Fig. 10 Result of experiment-2 (receiving time and throughput).

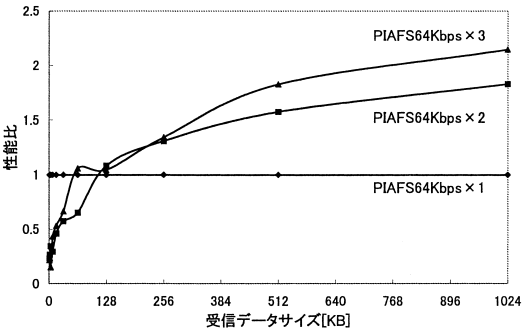

図 11 実験 2 の性能比
Fig. 11 Result of experiment-2 (ratio).

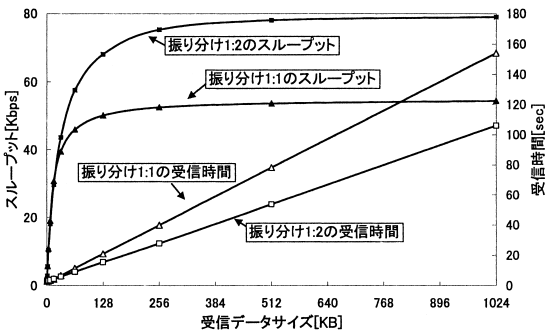

図 12 振り分け比率の違いによる影響
Fig. 12 Result of experiment-3.

クラスタ内は高速な LAN で構成されているため TCP の 3WAY ハンドシェイクに要する時間は無視できるが，各ホストと Web サーバ間の TCP 接続においては無視できない時間となる．今回実験に使用した Web サーバと AP はともに大学内の LAN 上にあるものを使用したため比較的早く接続確立をできたが，小さいサイズのファイルを受信するときには影響がある．また WebSHAKE では受信するファイルのサイズ情報を取得するためにまず HEAD 要求を送信するが，このときの接続確立もオーバヘッドの要因の 1 つになっている．

4.4.2 実験 2 の結果と評価

実験 2 の結果を**図 10**，**図 11** に示す．この図より各無線リンクのスループットにばらつきがある場合でも，実験 1 の結果と同様に大きなサイズでは平均スループットが向上していることが分かる．1 Mbyte のファイルを受信した場合，2 本の無線リンクで受信したときは約 1.8 倍，3 本の回線で受信したときは約 2.2 倍の性能が得られた．

理想的には n 本のリンクを共有すれば性能は n 倍であることが望ましいが，実験 2 の環境では十分な性能向上が認められなかった．今回使用した外部への無線リンクは図 7 で示した 3 本であり，各リンクのスループットにはばらつきがある．特に PIAFS64K(2)+AP3 の組合せからなる無線リンクは図 7 から分かるように 40 Kbps 程度のスループットしか得られていない．リンク 3 本を共有したときに性能向上が 2.2 倍にとどまった原因としてこのリンクが全体の性能の足かせとなっていること考えられる．

速度の遅い無線リンクが全体のスループットに影響を及ぼす理由は，SHP によるファイルの再整列・結合処理にある．WebSHAKE では 1 つのファイルを複数に分割して受信するため，最終的には分割したデータを再整列・結合処理しなければならない．結合処理では分割したデータに付加されているヘッダを取り除き，HEAD 要求で受信したヘッダを先頭に付加して結合する必要がある．このとき分割したデータすべてが届かないと結合できないため，スループットは一番遅いリンクに依存することになり，HTTP クライアントへ結合したファイルを送信できない．

4.4.3 実験 3 の結果と評価

実験 3 の結果を**図 12** に示す．この図から異なった帯域幅を持つ回線には単純に等分配するのではなく，

帯域を有効に利用できるように分配率を決めた方が有効であることが読み取れる．同じ数の無線リンクを共有していても，データを単純に等分配するか無線リンクの実効帯域などに応じて振り分けるかによって性能比として約 1.5 倍の開きがある．これは異なる帯域のリンクに同じサイズのデータを割り当てると速度の速いリンクは受信を完了し，遅いリンクが受信し終わるまで待つことになるためである．性能の違いは受信するファイルのサイズが大きいほど顕著に現れている．

4.5 考　　察

WebSHAKE では，n 本のリンクの帯域幅がすべて同じであるとしても受信中に実効帯域の変動があり，そのうちの 1 本の実効帯域が十分に得られない場合，他の $n-1$ 本からの受信を完了し，速度の遅いリンクの受信待ち状態となる．今回の実験でも 4.4.1 項にあげた 2 つのオーバヘッド以外に，この受信待ちも大きなスループットの低下の原因であることは実験 2 の結果より明らかである．WebSHAKE における最適な振り分け比率とは，共有している通信回線が同時にデータを受信し終わるときの比率であると考えることができる．もちろん完全に同時に受信を終了するような時間を算出することは困難であるが，できるだけすべての回線を有効活用し，いずれかの回線が使用されていないような無駄な時間を極力少なくすることが WebSHAKE では重要となる．

また理論的な解析結果と実験 1，2 の結果から，Web SHAKE の特性として大きなサイズのファイルを受信するときは十分な通信速度向上が得られるが，数 Kbytes 程度の小さいファイルを受信するとかえって性能が悪化することが確かめられた．この性能低下を防ぐための方法として，SHP による最初の HEAD 要求を Partial GET とする方法がある．Partial GET では，ファイルの一部分のデータだけでなくファイル全体のサイズを知ることができる．このため，HEAD 要求の代わりに Partial GET を用い，ファイルの先頭のデータを一部受信するようにすれば，小さいファイルは一度の SHP-Web サーバ間のメッセージ交換でファイル全体を受信できる．サイズの大きなファイルの場合は，最初の Partial GET で受信できなかった残りのデータを分割して複数のリンクを同時に利用して受信すればよい．

本稿の実験では，1 つのファイルを HTTP で受信し受信するファイルサイズを変動させたときのスループットしか計測していない．しかしながら現実の HTML 文書には画像などが複数含まれているので，クライアントはサイズが異なる複数のファイル（HTML 文書や画像データなど）を一度に受信する必要がある．Web ブラウザは受信した HTML 文書を走査した後，埋め込まれている画像データなどを受信する．このためサムネイル化された画像など小さなファイルサイズのデータが複数埋め込まれている場合，Web ブラウザは SHP に同時に複数の TCP コネクションを張ることになる．このように小さいサイズのファイルが複数存在するときに 1 つのファイルに対して複数に振り分けるとかえって性能が悪化することが予測される．

この問題に対処するため，GET 要求自体を各端末に振り分ける手法も SHP には実装されている．この手法は図 2 の①で SHP が Web クライアントから複数の GET 要求を受信すると，それらを他の端末へも送信する．すなわち 1 つのファイルを複数の端末で受信するのではなく，複数のファイルを複数の端末で受信する．今回の実験でもファイルサイズが小さい場合，振り分けることでのオーバヘッドが確認されているので，この手法はそのような場合に有効であると予測される．

5. おわりに

通信回線共有方式を用いた高速 Web アクセス方式 WebSHAKE を提案し，それを実現するソフトウェアとして SHAKE HTTP Proxy（SHP）を実装した．またこれを用いた実験により，本提案方式で Web からのファイル転送速度の向上が図れることを確認した．

SHAKE で想定するクラスタは一時的に回線を共有するものどうしのグループであり，つねに参加，脱退を繰り返すアドホックネットワークである．今後，複数の無線リンクを有効活用するための動的な振り分け制御手法，クラスタの管理手法について検討する予定である．また 1 つのファイルを分散して受信する手法と，複数のファイルを分散させて受信する手法をハイブリッド化し，自動的に最適化が行う機構についても検討する予定である．

参 考 文 献

1) Mineno, H., et al.: Multiple paths protocol for a cluster type network, *Int. J. Commun. Syst*, Vol.12, pp.391–403 (Dec. 1999).
2) Sklower, K., Lioyd, B., McGregor, G. and Carr, D.: The PPP Multilink Protocol (MP), RFC1990 (Aug. 1996).
3) 神尾享秀，児島史秀，藤瀬雅行：384 kbps-PHS 実験装置の概要と性能評価，情報処理学会研究報告，Vol.99, No.80, pp.33–40, 99-MBL-10 (1999).

4) Fielding, R. and Frystyk, H.: Hypertext Transfer Protocol — HTTP1.0, RFC1945 (May 1996).
5) Fielding, R., et al.: Hypertext Transfer Protocol — HTTP1.1, RFC2616 (June 1999).

(平成 13 年 6 月 6 日受付)
(平成 14 年 3 月 14 日採録)

推 薦 文

本稿は，個々の移動端末が持つ無線回線を複数のホストによってアドホック的に共有することにより，高速・高品質な移動通信環境を実現することを目的としたものである．本稿では，筆者らによって提案されている通信回線共有方式について，Web アクセスを対象として任意の WWW サーバとの通信を可能とする改良方式を提案し，その実現方法やプロトタイプの実装・評価に関しても詳細に記述している点が評価できる．（MBL 研究会運営委員 寺岡文男）

小西 洋祐（学生会員）

昭和 53 年生．平成 13 年静岡大学情報学部情報科学科卒業．現在同大学大学院情報学研究科博士前期課程（情報学専攻）に在学中．コンピュータネットワーク，通信プロトコルに関する研究に従事．

石原 進（正会員）

昭和 47 年生．平成 6 年名古屋大学工学部電気学科卒業．平成 11 年同大学大学院工学研究科博士後期課程電子情報学専攻修了．平成 10 年度日本学術振興会特別研究員．平成 11 年静岡大学情報学部情報科学科助手．平成 13 年同大学工学部システム工学科助教授．博士（工学）．モバイルコンピューティング，無線環境用 TCP/IP に関する研究に従事．電子情報通信学会，ACM 各会員．平成 9 年度電気通信普及財団賞．

水野 忠則（正会員）

昭和 20 年生．昭和 43 年名古屋工業大学経営工学科卒業．同年三菱電機（株）入社．平成 5 年静岡大学工学部情報知識工学科教授，現在，同大学情報学部情報科学科教授．工学博士．情報ネットワーク，プロトコル工学，モバイルコンピューティングに関する研究に従事．著書としては「プロトコル言語」（カットシステム），「コンピュータネットワーク概論」（ピアソン・エデュケーション）等がある．電子情報通信学会，IEEE，ACM 各会員．当会フェロー．