

通信回線共有方式を利用した高速 Web アクセスの実現

小西 洋祐[†] 橋本豊大[‡] 石原 進[†] 水野忠則[†]

静岡大学情報学部[†] 静岡大学大学院理工学研究科[‡]

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[†], Toyohiro Hashimoto[‡], 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ー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ー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、多様なニーズに応えるため画像や音声、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今だ十分とは言い難い。また無線環境では TCP のハンドオフやビットエラーに伴う性能の悪化などにより、さらにデータのダウンロードに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

[†]Faculty of Information, Shizuoka University

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

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WebSHAKEを提案する。WebSHAKE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。

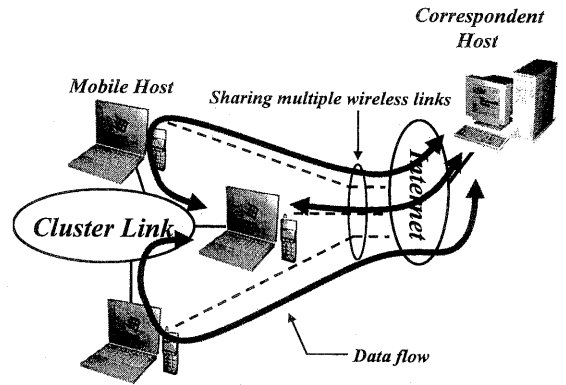


図1 通信回線共有方式 (SHAKE) による通信例

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してホームページ閲覧を楽しむなどという時には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も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、それ自身がも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。この方式を我々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。

- 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時、または利用不可能な時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ーフェイ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とは異なる。この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]。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関係を必要とする。例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を利用する上では、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。WebSHAKE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]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サーバとして振舞うので、受信した分割ファイルはそのままGET要求を発生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理・結合し、HTTPクライアントへ送信する。この時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。

HEAD要求やPartialGET要求、各クラスタの管理、およびファイルの再整理・結合処理はすべてSHPが行う。WebSHAKEではSHPがPartialGET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う必要はない。またSHPがHTTP Proxyサーバとして振舞うので、クラスタ内の移動端末でもSHAKE用の特別なHTTPクライアント等を用意する必要がない。次節ではSHPについて詳しく説明する。

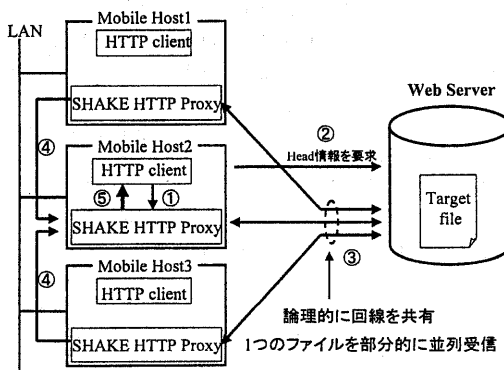


図2:WebSHAKEアーキテクチャ

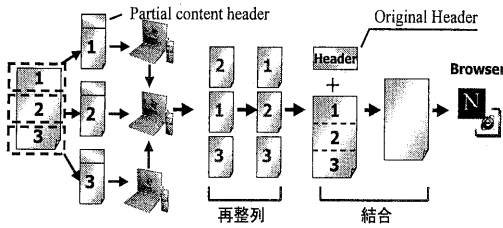


図3:WebSHAKEにおけるファイルの再整理・結合

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来るので、SHPは常に流れるデータの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 ファイルの再整理・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。例えば1000[bytes]のファイルの要求後、0-500[bytes]をHOST1、501-1000[bytes]をHost2 経由で受信した時、Host2の通信速度が速い時は、501-1000[bytes]が最初に受信される時もある。したがってSHPはこれら分割ファイルが誤った順序で受信されても、正しい順序に整理する。

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。また全てのMHでSHPが動作するものとする。ここではTCPコネクション確立時のオーバー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。

$$T_n = 2d_w + \frac{2H + D}{B_w} + \tau_s \quad (1)$$

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。

$$T_s = (5\tau_p + 2\tau_s) + (2d_h + 4d_w + 2d_c) + (2H + D)/B_h + (4H + D/n)/B_w + (2H + D/n)/B_c \quad (2)$$

クラスタは高速リンクで構成することを前提としているので、 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(2)式は以下のように書き換えることができる。

$$T_s = (4d_w + 2d_c) + (2\tau_s + 5\tau_p) + \frac{4H + D/n}{B_w} + \frac{2H + D/n}{B_c} \quad (3)$$

簡略化のため詳細なステップは省くが、通常の 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 要求を直接送る。このときのオーバーヘッドが $\tau_p + 2d_w + H/B_w$ となる。
2. 1 つのファイルをクラスタ内の各移動端末が並列に受信するのに $(2H + D/n)/B_w$ を要する。
3. SHP は全ての部分データを整列し、ヘッダを再構成してクライアントに返送する。ここで SHP の処理時間とクライアントへの転送時間の和 $\tau_p + d_h + (H + D)/B_h$ を要する。

4.1.3 WebSHAKE と通常の Web アクセスとの比較

応答時間の比 T_n/T_s を計算すると以下のような式となる。ただし $B_w \ll B_c$, $d_c \approx 0$ とする。

$$T_n/T_s = \frac{n + 2nH/D + n(2d_w + \tau_s)B_w/D}{1 + 4nH/D + n(4d_w + 2\tau_s + 5\tau_p)B_w/D} \quad (4)$$

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_n/T_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ー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。

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 サーバとして振るまうように見えるので、一般の Web ブラウザからプロキシとして選択して利用することが可能である。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),

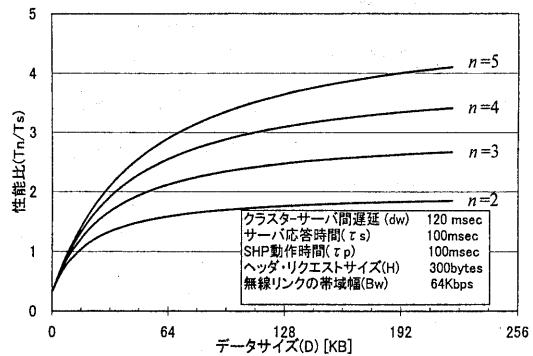


図4:理論的な WebSHAKE の性能比

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 PC 3 台を使用した。ノート PC には Sharp の PJ2-X3 (CPU: PentiumII 300Mhz, Memory: 128 Mbytes)1 台と Panasonic の CF-M2 (CPU: Cereron 500Mhz, 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無く、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 本使用した時と、2 本共有した時で、Web サーバから 1Kbytes~1Mbytes までの様々なサイズのファイルを受信した時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。図5の通り、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

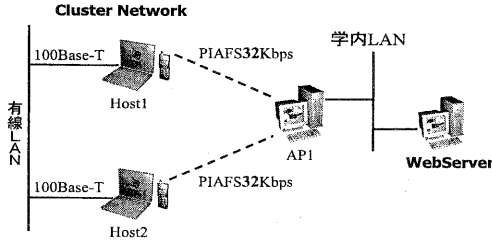


図 5 : 実験 1 の環境

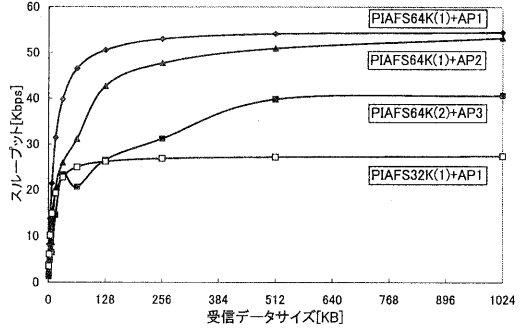


図 7 : 実験に使用した通信経路の性能

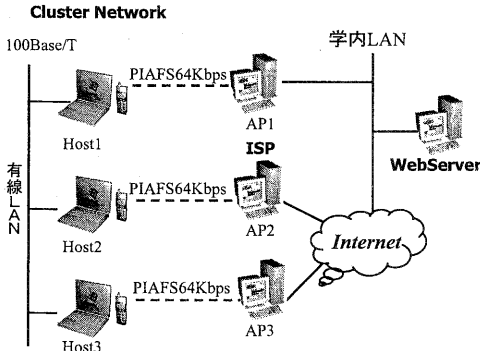


図 6 : 実験 2 の環境

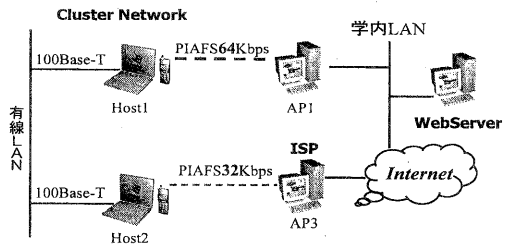


図 8 : 実験 3 の実験環境

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 64Kbps の帯域幅を持つ無線リンクを 1~3 本共有した時の 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時の受信時間を測定した。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 1Kbytes~1Mbytes のデータを HTTP で受信した時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 PIAFS64K(1)+AP1 と PIAFS64K(1)+AP2 の 2 種類である。1 本のときの測定に使用した無線リンクは 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。

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 32Kbps であり、もう一つは 64Kbps 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、64Kbps の回線には 32Kbps 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 つの条件でファイルを受信するまでの時間を測定した。

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程スループットが向上しており、1Mbytes のファイルを 2 本の無線リンクで受信した時は、1 本で受信した時と比べて約 1.85 倍の性能向上が確かめられた。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、性能も予測した通りの結果が得られた。

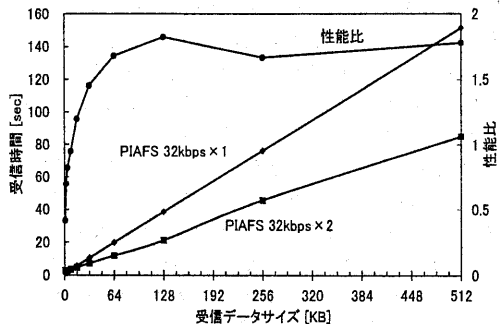


図 9 : 学内の AP を用いた PIAFS32Kbps × 2 の実験結果

理論的な予測通り大きなファイルを受信する時には性能の向上が見られる反面、小さなファイルを受信している時には性能の低下がみられる。図 9 から 16Kbytes 以下のデータを受信した時には性能が無線リンク 1 本を使用した時よりも後退していることがわかる。この理由としては 4.1 節で検討したオーバーヘッドの他に 4.1.2 では無視していた TCP コネクション確立に伴うオーバーヘッドが考えられる。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、各ホストから Web サーバの TCP コネクションでは無視できない時間となる。今回実験に使用した Web サーバは大学内にあり、AP も同じ学内の 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時には影響がある。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。この時のコネクション確立もオーバーヘッドの要因の 1 つになっていると思われる。

現在の実装では TCP コネクションの確立はブロッキングモードで行っているため、TCP の確立が成功するかまたはタイムアウトになるまで処理が戻って来ないという問題がある。このコネクション確立を非ブロッキングモードにすることで多少は改善されると予測できる。

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は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わかる。1Mbyte 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

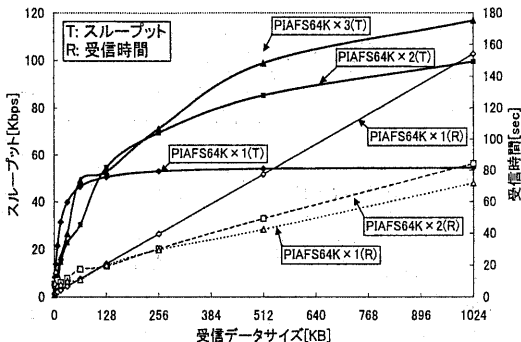


図 10 : 通信回線を 1~3 本共有しての実験結果

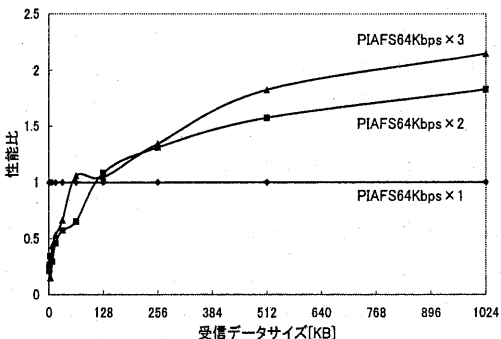


図 11 実験 2 の性能比

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 40Kbps 程度のスループットしか得られていない。リンク 3 本を共有した時に性能向上が 2.2 倍に留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理・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理・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カットし

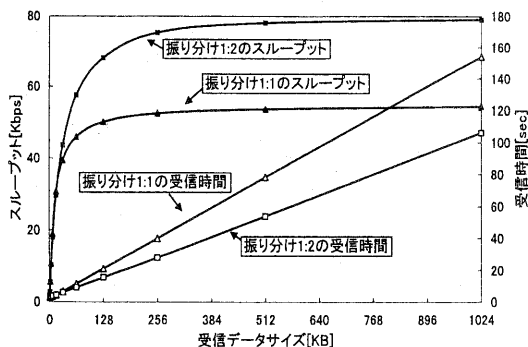


図 12 振り分け比率の違いによる影響

て、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。この時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ほう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

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 $n-1$ 本は受信を完了し、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 に挙げた 2 つのオーバーヘッド以外に、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 の結果を見れば明白である。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る時の比率である。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、できるだけ全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 WebSHAKE 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返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 を使い、ファイルの先頭のデータを一部受信するにすれば、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。

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、それを実現するものとして SHP を実装した。またこれを用いた実験により本方式で Web からのファイル転送速度の向上が計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同士のグループであり、常に参加、脱退を繰り返すアドホックネットワークである。今後の課題としてクラスタを動的に管理し、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について検討する予定である。

参考文献

- [1] H. Mineno, et. al., "Multiple paths protocol for a cluster type network," Int. J. Commun. Syst, Vol. 12, pp. 391 - 403, December 1999
- [2] K. Sklower, B. Liloyd, G. McGregor, D. Carr "The PPP Multilink Protocol (MP)," RFC1990, August 1996
- [3] 神尾亨秀, 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.
- [4] R. Fielding, et. al., "Hypertext Transfer Protocol - HTTP1.1," RFC2616, June 1999.