

通信回線共有方式のためのクラスタ資源発見・管理機構の実装

小西洋祐^{†1} 石原進^{†2} 峰野博史^{†3}
太田賢^{†4} 水野忠則^{†3}

近年 2G, 3G 携帯電話網, PHS 網, 無線 LAN ホットスポット等, 様々な通信インフラが整備され, また端末側も複数のネットワークインターフェイス (NetI/F) を備えている. 筆者らはこのような多種のネットワークが混在したヘテロジニアスな環境下において, 複数の端末が一時的なネットワーク (クラスタ) を構築し, クラスタ内において各端末が協調してそれらの NetI/F 資源を共有することで, クラスタとクラスタ外との間の効率的な通信を実現する方式: 通信回線共有方式 SHAKE を提案している. SHAKE におけるクラスタは時, 場所に依存せず柔軟に構築することが可能なアドホックネットワークを想定しているため, 共有する NetI/F 資源を効率よく利用するためには, 常に変動, 増減するこれらの資源を自律的に発見・管理する機構が望まれる.

本稿では, 共有する NetI/F 資源を自律的に発見・管理し, 動的にクラスタを構築する機構として SCCM (SHAKE Cluster Control Manager) の実装について述べる. またこれまでに実装・評価した HTTP 専用型の SHAKE の実現である Web SHAKE システムに SCCM を適用し, 共有する NetI/F 資源が変動・競合した状況における評価を行う.

Implementation of Cluster Resources Management System for Multiple Wireless Links Sharing System

YOUSUKE KONISHI,^{†1} SUSUMU ISHIHARA,^{†2} HIROSHI MINENO,^{†3}
KEN OHTA^{†4} and TADANORI MIZUNO^{†3}

Nowadays mobile terminals have multiple network interfaces such as WirelessLAN, IrDA, Bluetooth, and PHS. To offer high speed communication on wireless links, we have proposed SHAKE (SHARing multiple paths procedure for cluster network Environment). In SHAKE, we assume that cluster network is composed ad-hocery. The cluster network is dynamic and mobile hosts may join to the cluster and leave at any time. We have proposed SCCM (SHAKE Cluster Control Manager) that configures a the cluster is dynamically realized by discovering and advertising shared network resources informations autonomously. The SCCM enables mobile terminals to consist of the Cluster Network dynamically.

We implemented the SCCM and applied it to Web SHAKE, which is an implementation of SHAKE and works only on HTTP. In addition, we implemented a method for dynamic traffic and priority control for Web SHAKE by using the SCCM. We evaluated the performance of the system by experiments.

1. はじめに

近年 2G, 3G 携帯電話網, PHS 網, 無線 LAN ホットスポット等, 様々な通信インフラが整備され, 時, 場所等に依存せずインターネットに接続することが可能になってきた. またそれに伴い, 携帯端末は複数のネットワー

クインターフェイス (有線 LAN, 無線 LAN, PHS, 携帯電話, Bluetooth, IrDA 等) を備えていることが多くなったが, ユーザはこれら複数のネットワークインターフェイス (以下 NetI/F) のいずれかを状況やユーザの嗜好に応じて使い分けて利用しているのが現状である. しかしながら今後ますます進むと予測される多種混在のヘテロジニアスなネットワーク環境においては, これら環境の変化を隠ぺいし, より柔軟で快適な通信ができることが期待される.

そこで筆者らは, 複数の携帯端末が一時的にネットワークを構築し, それぞれが持つ外部への NetI/F を共有することで高速・高品質なデータ通信を実現する方式: 通信回線共有方式 SHAKE (SHARing multiple paths procedure for cluster network Environment) を提案している¹⁾. ここで NetI/F 資源を共有することを目的として構成された一時的なネットワークをクラスタと呼んで

^{†1} 静岡大学大学院 情報学研究科
Graduate School of Information, Shizuoka University
y-konisi@mizulab.net

^{†2} 静岡大学工学部 システム工学科
Faculty of Engineering, Shizuoka University
ishihara@ishilab.net

^{†3} 静岡大学 情報学部
Faculty of Information, Shizuoka University
{mineno,mizuno}@mizulab.net

^{†4} NTT ドコモ マルチメディア研究所
Multimedia Laboratories, NTT DoCoMo, Inc.
ken@mml.yrp.nttdocomo.co.jp

いる。

ヘテロジニアスなネットワーク環境下で、移動するユーザが通信回線共有方式を使用するためには、端末が自律的に協調して共有可能な NetI/F 資源の発見・管理を行い、クラスタの構築を行えることが必要である。筆者らはこれまでに各携帯端末が共有する資源情報を、各端末が自律的に告知・発見を行うことで、動的にクラスタを構築、共有する資源の発見・管理を実現するための機構として、SCCM(SHAKE Cluster Control Manager)の提案・検討を行ってきた²⁾。本稿では、この SCCM の実装について述べる。またこれまで共有する NetI/F が変動しない状況しか取り扱うことができなかった SHAKE の実装の 1 つ Web SHAKE³⁾ に SCCM を適用することで、その有効性を評価する。さらに複数のユーザにおけるクラスタ内の共有資源の競合時についても評価を行う。

以下本章の構成について述べる。2 章では通信回線共有方式 SHAKE の概要について説明し、共有資源の発見・管理の必要性と課題について明らかにする。3 章では、クラスタ資源管理機構：SCCM について説明し、4 章でその実装手法について述べる。また 5 章で、Web SHAKE への SCCM 適用例について説明する。6 章では、SCCM を適用した Web SHAKE において利用可能な NetI/F が増減した場合や資源競合が発生した時の状況において性能評価を行い、7 章でまとめとする。

2. 通信回線共有方式 SHAKE

2.1 概 要

複数の NetI/F をもつ複数の移動端末が一時的にネットワークを構築し（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 NetI/F 以外にクラスタ内の他の移動端末の持つ NetI/F も利用すると、単体では低速・低品質な NetI/F 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 の主な利点として以下が挙げられる。

1. トラフィック分配による通信速度の向上、および異なる経路に冗長なデータを流すことによる信頼性の向上
2. 自身が外部への NetI/F を持たない時、または利用不可能な時でも他の端末の NetI/F を利用可能

複数の NetI/F を同時に利用する関連研究として PPP マルチリンク⁴⁾ や専用装置を使用した複数 PHS の同時利用方式⁵⁾、複数経路を用いたデータ転送方式⁶⁾ などがある。これらは個々の端末が持つ複数の NetI/F を同時に利用することでデータ転送レートの高速化を実現するが、端末が備える複数の NetI/F は固定的であり、動的に増減するような状況は想定していない。複数の端末が持つ NetI/F を共有する研究として、DICOS⁷⁾ がある。これはインターネットカー内での使用を想定し、車内に

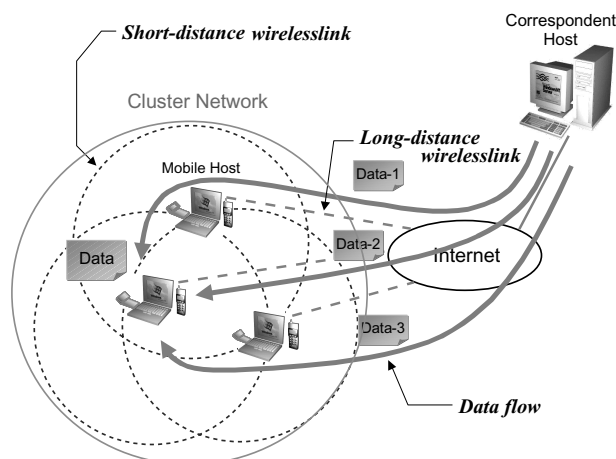


図 1 通信回線共有方式（SHAKE）を利用した通信例

において外部への接続を持つ複数の端末の NetI/F を、他のそれらを持たない端末がそれらを選択的に利用しインターネット接続性を共有する。ただし接続性を共有する車内は比較的静的であり、車外への NetI/F の性質の変動、車中ネットワークの構成は頻繁には変動しない。

SHAKE においては、個々の端末だけでなく他の端末がもつ NetI/F を同時に利用するため、動的な NetI/F の増減に対応する必要がある。また複数のユーザが NetI/F 資源の同時利用による資源の占有・競合についても考える必要がある。

2.2 クラスタ資源管理の必要性

ヘテロジニアスな環境下で、移動ユーザが SHAKE を利用することを想定すると、クラスタへの端末の参加・脱退、各端末の持つ NetI/F の性能の変化が頻繁に起こることが予測される。このため SHAKE を利用する上で効率よく資源を共有するためにはこれらを発見し、管理する必要がある。

またユーザの要求やアプリケーションによっては、SHAKE を使う利点は転送レートの向上の他にも存在する。例えば、あるユーザが重要な会話を Chat により行っていたと仮定する。この場合ユーザが求めるのは転送レートの向上ではなく、接続（セッション）の維持であると予測される。また Web の閲覧などにおいてもユーザの要求によっては、セキュアな通信路の選択を望む場合もある。このため、ユーザ要求、アプリケーション要求に適した経路を選択するためには、ユーザが各アプリケーションに求める要求、そして各アプリケーションごとに異なる性質に応じて SHAKE で利用可能な NetI/F 資源を適切に選び出す機構が望まれる。

3. クラスタ資源管理機構：SCCM

3.1 機 能

SHAKE 利用のために、クラスタを構成する端末は、それぞれ資源（NetI/F, CPU, バッテリなど）やその状態が異なる。そのため SHAKE 利用時には、各端末同士が

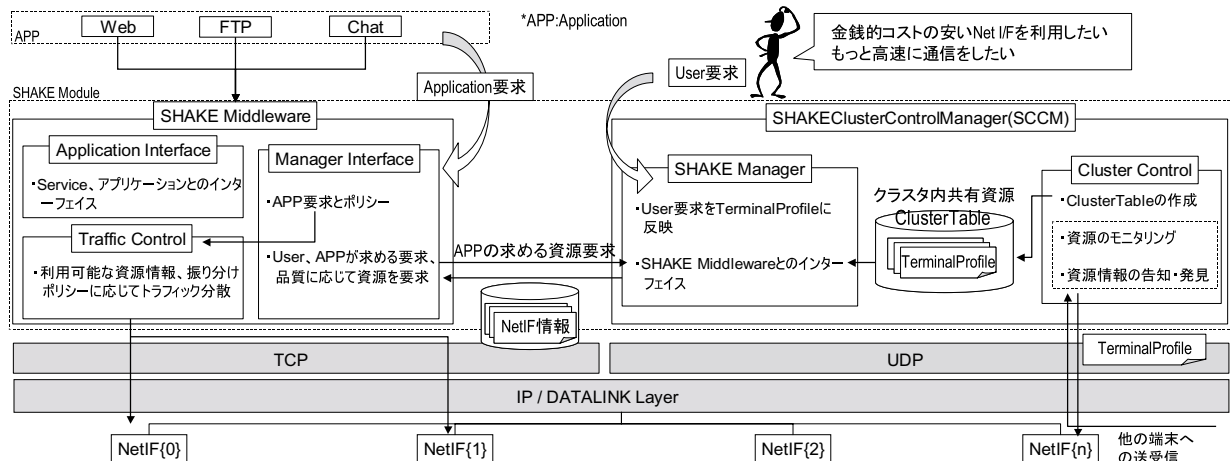


図 2 クラスタ資源管理部分を分離したシステム構成図

自律的に共有可能な資源情報を取得する必要がある。これを実現する機構として筆者らはこれまでに以下 4 つの機能を提供する SCCM (SHAKE Cluster Control Manager) の提案を行った²⁾。

- (1) 端末が持つ資源のモニタリング
- (2) 共有可能な資源の告知・発見
- (3) 共有資源の管理
- (4) 他のアプリケーションとのインターフェイス

SCCM は、各端末が持つ NetI/F 資源の変化を常時モニタリングし、定期的にこれらの情報を各端末に告知する。また他の端末が告知した情報を取得し、利用可能な共有資源の発見・管理を行う。SCCM は他のアプリケーションとのインターフェイスを持ち、各端末上で動作するアプリケーションは SCCM に要求することで共有 NetI/F 資源を取得することが可能である。

3.2 設 計

SCCM は複数のアプリケーションから利用できるように、他のアプリケーションとは独立して動作するよう設計されるべきである。図 2 に SCCM のシステム構成図を示す。SCCM は SHAKE を利用する各端末上のローカルで動作し、クラスタを構成する端末および資源情報の発見、SHAKE に利用可能な資源情報の告知、および SHAKE ミドルウェア (以下 SHAKE MID) へ資源情報を提供する。SCCM は Cluster Control と SHAKE Manager の 2 つの機能から構成される。

3.2.1 Cluster Control の設計

Cluster Control では (1) 端末が持つ資源のモニタリング (2) 共有可能な資源の告知・発見 (3) 共有資源の管理を行う。SHAKE では、インターネットに接続されている外部リンク、そしてクラスタを構成するための内部への無線リンク (以下クラスタリンク) の 2 つ特性を持った NetI/F を有する。外部への接続は各ユーザが DHCP などを利用し動的に IP および経路を取得し接続することが可能である。しかしクラスタ内において各端末は任意のタイミングでクラスタに参加、脱退するため、

DHCP のように固定サーバを設置し、集中的に管理することは適切ではない。そこでクラスタ内通信においては、各端末がクラスタリンクを用いて定期的に資源情報をブロードキャストすることで各端末のもつ資源情報の告知を行う。

実際の環境では、ユーザの意図しないリンクの切断等が起こりうるため、クラスタの脱退に際して、各端末は脱退を意味するメッセージを告知する必要はないものとした。各端末はそれぞれ独自の間隔で定期的に資源情報を交換している。この間隔の 2 倍を有効期限と定め、その期限を超過しても資源情報が得られないときは、その端末がクラスタから脱退したものとし、該当する資源情報を無効にする。

3.2.2 SHAKE Manager の設計

SHAKE Manager は (4) SHAKE MID とのインターフェイスを提供する。クラスタを構成する各端末では、SCCM と SHAKE MID が動作しているものとする。SHAKE MID はクラスタ内における SHAKE で利用可能な資源情報を保持しないため、SCCM に必要な情報を取得するための要求を行う。SCCM は、SHAKE MID からの要求に応じて、クラスタ資源情報が記述された ClusterTable から参照して応答する。SHAKE MID が要求する情報としては、現在共有可能な NetI/F 資源の一部またはすべての提供、最もセキュアな NetI/F 資源の提供などが考えられる。

4. 実 装

SCCM を構成するモジュールである Cluster Control と SHAKE Manager を Linux 上で実装した。以下それぞれについて説明する。

4.1 Cluster Control の実装

複数の端末で共有可能な資源としては CPU、メモリ、ストレージ、NetI/F、バッテリー等が考えられる。SHAKE では複数の NetI/F 資源を共有し、他のユーザが持つ CPU、バッテリーを消費するため、これらの情報を Cluster Con-

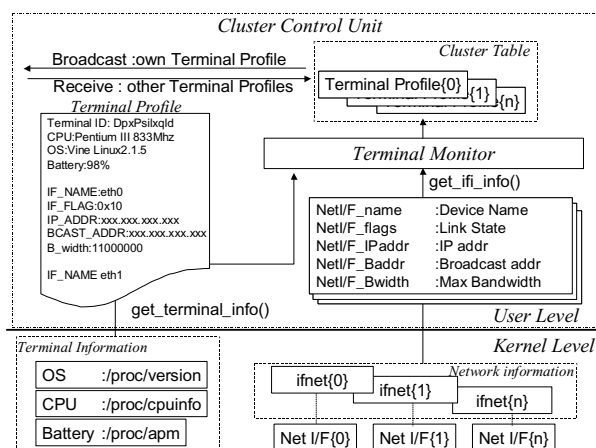


図 3 Cluster Control の動作

trol でモニタリングし、他の端末へ告知する。

図 3 に Cluster Control の動作を示す。端末内の資源モニタリングは Terminal Monitor にて定期的に行っている。OS, CPU, バッテリ, NetI/F 情報は Kernel から直接取得する。NetI/F の情報としてはカーネル内の ifnet 構造体に連結リストとして格納されている以下の情報を取得する。

- NetI/F_name: デバイスの名前
- NetI/F_flags: リンク状態フラグ
- NetI/F_IPaddr: IP アドレス
- NetI/F_Baddr: ブロードキャストアドレス
- NetI/F_Bwidth: 帯域幅

Terminal Monitor によって取得した端末の資源情報は、NetI/F_Baddr を用いてブロードキャストを行うと同時に、SCCM 内にて定義した Terminal Profile に格納される。また他の端末から受信した資源情報も同様に Terminal Profile へと格納する。図 4 に Terminal Profile の構成を示す。Terminal_ID はクラスタを構成する各端末同士をそれぞれ識別するために用い、クラスタ内で唯一の値を持つよう設定する。Terminal Profile は各端末でそれぞれ Cluster Table に格納、管理され、Terminal_ID で識別される。Terminal_ID ごとに管理しているのが Cluster Table である。Recv_Profile_TIME には他の端末から受信した Terminal Profile の受信時刻を格納する。各 Terminal Profile ごとに受信した時刻を記録し、それらの受信間隔の 2 倍の時間をその有効期限と定め、Profile_TTL に格納する。この時間を超えた場合にその Terminal Profile を Cluster Table から削除することでクラスタからの脱退を実現する。また一度登録された端末への RTT を計測し、Clusterlink_RTT へ格納する。

4.2 SHAKE Manager の実装

SHAKE Manager と他の SHAKE MID とのインターフェイスは、UDP を用いたプロセス間通信において実現している。SHAKE Manager は SHAKE MID に対するサーバとして機能し、SHAKE MID から受信した要求に

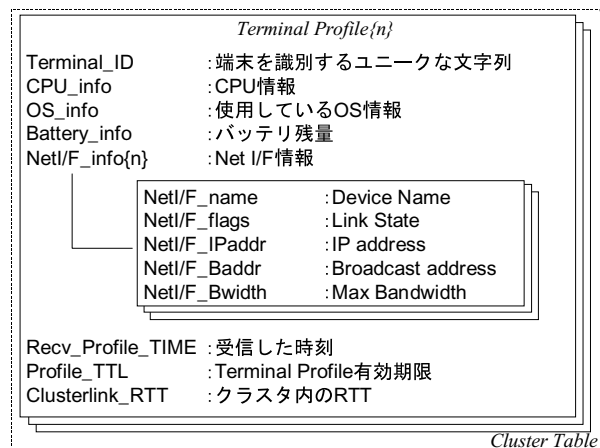


図 4 TerminalProfile と Cluster Table

応じた共有資源情報を Cluster Table から参照し応答を返す。外部 NetI/F を複数の端末で共有するためには、クラスタ内リンクと外部へのリンクの最低 2 つの NetI/F に関する情報が必要である。そのための要求を定義したこの要求を受け取った SHAKE Manager は ClusterTable の中から該当する Terminal Profile 情報のみを返す。

5. SCCM の適用例：Web SHAKE

SCCM による動的な資源管理機構を実現する例として、これまで実装・評価をしてきた Web SHAKE³⁾ への適用を行った。Web SHAKE は利用プロトコルを HTTP 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 を利用した通信が可能なアプリケーションの一つである。Web SHAKE では、クラスタ内の各移動端末で SHAKE の機能を組み込んだ SHAKE Proxy(以下 SHP) を動作させることで複数経路へのトラフィック分散を実現する。SCCM 適用前の Web SHAKE はクラスタ資源管理を SHP 内に静的に保持していたため、共有する NetI/F の増減を取り扱うことができなかった。SCCM を適用することにより、このような状況を取り扱うことが可能となる。

5.1 システム設計

Web SHAKE と SCCM のシステム構成を図 5 に示す。これまで SHP 内において静的に管理されていたクラスタ資源の管理は SCCM が行う。SHP は SCCM Interface を用いて SCCM の SHAKE Manager に SHAKE で利用可能な NetI/F 資源情報を要求し、これを取得する。以下 Web SHAKE と SCCM 間のメッセージのやり取りについて説明する。

- (1) SHP はローカルホストで動作する HTTP Client (Web ブラウザ) から GET 要求を受信し、同時に HEAD 要求を Web サーバへ送信することで受信ファイルサイズを取得する。
- (2) SHP は HTTP Client から GET 要求を受信した時点で、SCCM Interface を通じて現在利用可能

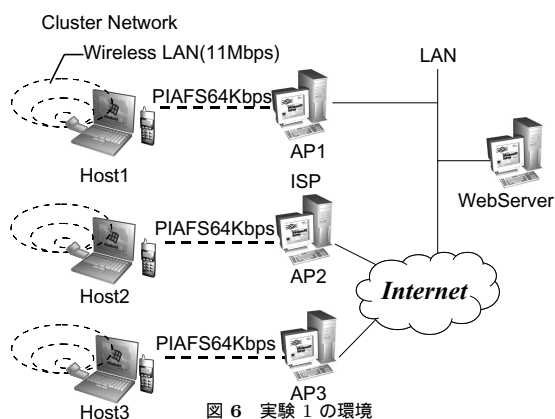


図 6 実験 1 の環境

SCCM を動作させた。

6.2 実験 1：利用可能な共有 NetI/F の増加・減少

6.2.1 実験内容

実験 1 では、要求分配前に利用可能な共有 NetI/F 資源が増減した時に、SCCM を利用することでその変化に対応することができるかを評価した。図 6 に実験環境を示す。Host1, Host2, Host3 が無線 LAN でクラスタを構成し、各ホストにおける外部 NetI/F の帯域はそれぞれ 64Kbps である。Host1 が SHAKE を利用する側 (HTTP Client から直接 GET 要求を受信する SHP を以下 Master SHP と呼ぶ) であり、Host2, Host3 は利用される側 (Master SHP から Partial GET 要求を受信する SHP を以下 Slave SHP と呼ぶ) である。

このような環境において、Web サーバから 5 つの画像ファイル (データサイズは全て 1Mbytes) のデータを受信するために、Host1 上で動作している HTTP Client で 5 回の GET 要求を順次発生させた。HTTP Client が GET 要求を送信するタイミングは、前の要求の応答として 1 つのファイルを全て受信した直後である。初回要求時には Host2, Host3 の外部 NetI/F は各アクセスポイント (以下 AP) には接続しない。2, 3 回目の GET 要求送信で順次 Host2, Host3 の外部 NetI/F を AP に接続し、共有可能な NetI/F 資源を増加させた。また 4, 5 回目の要求時には順次 Host3, Host2 と AP 間をそれぞれ切断し共有可能な NetI/F 資源を減少させた。このような実験内容において、Host1 が受信したデータサイズ、スループット、受信時間を測定した。

6.2.2 実験 1 の結果と考察

図 7 に実験 1 の結果を示す。これは初回要求発生時刻を 0 とした時の Host1 が受信したデータサイズ、スループットの推移を示している。1 つ目の要求発生時には Host2, 3 の NetI/F 資源を利用できないため、Host1 が持つ外部 NetI/F で受信を行っている。Master SHP は 2, 3 回目の要求発生時には Host2, Host3 の外部 NetI/F が利用可能状態であることを SCCM によって知ることができる。そして Master SHP は 5.2 節で示したトラフィック分配規則に従って要求データを分割し、複数の NetI/F

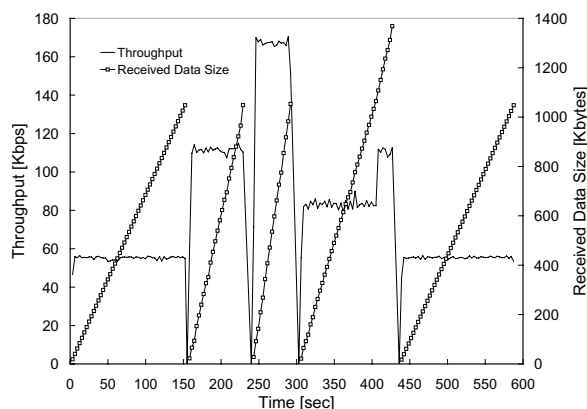


図 7 実験 1 の結果

資源を利用した並列受信を行う。2 回目の要求発生時には平均 110Kbps, 3 回目には平均 155Kbps のスループットが得られていることから SCCM によって NetI/F の増加に対応できていることが分かる。同様に 4, 5 回目の要求発生時には Host2, Host3 がそれぞれ外部 NetI/F を各 AP から切断するため、この変化を受けた SCCM は共有可能資源情報を変更し Master SHP へ通知する。Master SHP は共有可能な NetI/F 資源の減少を SCCM により取得するため、スループットも 2 回目、1 回目と同程度に戻っている。以上から SCCM が有効に動作し、SHP が動的な共有資源の変動に対応していることが確認できた。

ここで 4 回目の要求で、2 回目の時と比べスループットが減少しており、さらに Host1 が受信したデータサイズが 1.3Mbytes と他より多く受信している理由を考える。スループットが減少した理由は、Host2 が利用している PHS 通信がベストエフォート型であったため、データ受信途中に 64Kbps から 32Kbps へと利用可能帯域が減少したからと考えられる。また受信したデータサイズが 300KB 程度多い理由は、5.2 節のトラフィック分配により Host1 は全てのデータを受信する要求を Web サーバへ送信しているからである。つまり Host1 は、Host2 から分割したデータを受信するまでデータの読み込みを継続し、重複してデータの受信をしているからである。Host2 で利用可能な帯域が変動しなければ、Host2 でのデータ受信が早く終了し、Host1 で重複した受信はほとんど行われないと考えられる。

6.3 実験 2：データ通信途中の NetI/F 資源の減少

6.3.1 実験内容

実験 2 は、GET 要求を分配した後、データ受信中に要求を依頼された端末がクラスタから脱退 (利用可能な NetI/F 資源の減少) した場合の動作を検証した。図 8 に実験環境を示す。Host1 および Host2 の 2 台でクラスタを構成し、各ホストにおける外部 NetI/F の帯域はそれぞれ 32, 64Kbps である。Host1 が Master SHP であり、Host2 は Slave SHP として機能するようにした。Host1 で動作している HTTP Client は、Web サーバか

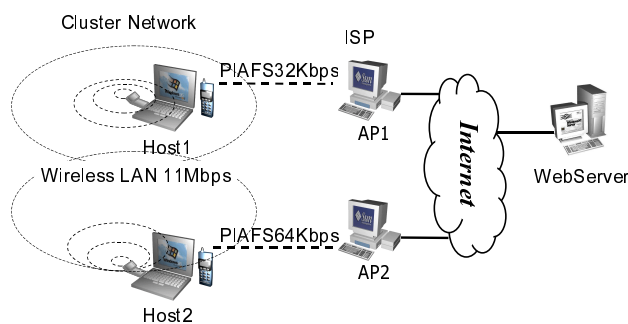


図 8 実験 2, 3 の環境

ら 1Mbytes のデータを受信するために、GET 要求を送信する。この時 Host1, Host2 共に外部 NetI/F を AP に接続しているが、Host1 がデータ受信時、要求を送信してから 30 秒後に Host2 の PHS 通信カードを AP から切断させた。このようにしてデータ通信途中の NetI/F 資源の減少時に、Host1 が受信したデータサイズ、スループット、受信時間を測定した。

6.3.2 実験 2 の結果と考察

図 9 にその実験での結果を示す。これは Host1 が持つ NetI/F, Host2 が持つ外部 NetI/F から受信したスループット、受信時間、受信したデータサイズの変化を示している。0-30 秒間は、Host1 は両端末が保持する NetI/F を用いてデータ受信を並列に行っている。しかし 30 秒経過し、Host2 の NetI/F が AP から切断され、Host2 の NetI/F が利用できなくなると、Host1 が持つ NetI/F のみでデータ受信を行っている。前述したように Host1 は全てのデータを受信するための要求を送信しているため、要求を依頼した他の端末が通信途中で切断しても Host1 のみで送受信を完了することができる。

実験 1 および実験 2 の結果で示された通り、Master SHP は全てのデータを受信するための要求を Web サーバへ送信しているため、Slave SHP に振り分けたデータが予期した通りに到着しない場合、Master SHP は冗長なデータを受信することになった（実験 1：300KB、実験 2：200KB）。

6.4 実験 3：資源競合時における性能

6.4.1 実験内容

実験 3 では、複数の端末が同時に、さらに通信途中に GET 要求を発生させた状況において、5.3 節で示した要求の抑制・優先制御が有効に動作するかを検証した。実験環境は図 6 と同じである。実験 1, 2 ではクラスタを構成する 1 つの端末のみが GET 要求を発生させたが、実験 3 では Host1, Host2 の双方で動作している HTTP Client で GET 要求を発生させた。発生させるタイミングは以下の 2 つである。各端末間の要求発生時刻の同期は NTP(Network Time Protocol) を利用した。

- (1) Host1, Host2 双方が同時に 1Mbytes のデータを受信するための GET 要求を発生

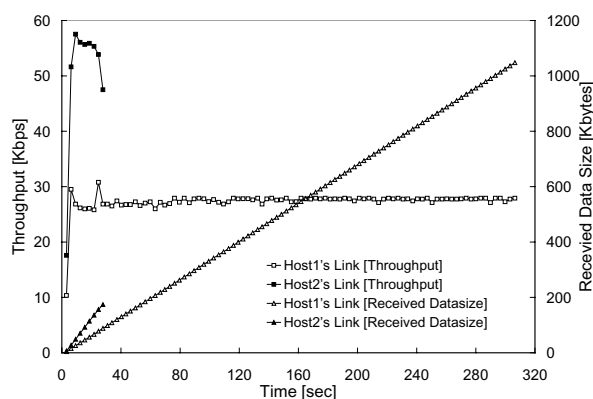


図 9 実験 2 の結果

- (2) Host1 が 1Mbytes のデータを受信するために GET 要求を発生させた後、60 秒後に Host2 が 512Kbytes のデータを受信するための GET 要求を発生

6.4.2 実験 3(1) の結果と考察

実験 3(1) の実験結果を図 10 に示す。これは Host1 におけるスループットおよび受信データサイズの変化を示したものである。図 10 より、0-155 秒間は Host1 が持つ PHS 通信カードによるデータ受信のみが行われており、Host2 が持つ PHS 通信カードは使用されていないことがわかる。これは、Host2 で動作している SHP において (4) (5.3 節図 5) の Priority control 部が機能しているためである。Host2 も Host1 と同時刻に GET 要求を受信しているため、お互いの端末上で動作している SHP は、それぞれが端末自身の上で動作するクライアントが要求したデータ通信を行っている。Host1, Host2 上の各 SHP は自身のデータ通信を優先し、終了するまで他の端末から受信した Partial GET 要求は Web サーバへ送信しない。

155 秒を経過した時点で Host2 の PHS 通信カードからデータ受信が始まる。Host2 は Host1 より 2 倍の帯域を持つ外部リンクを保持しているため Host1 よりも先にデータを受信し終える。このとき Host2 で動作している SHP は、それまで保留されていた優先度処理によって Host1 から受信した要求を Web サーバへ転送し、データの受信を開始する。

6.4.3 実験 3(2) の結果と考察

実験 3(2) の実験結果を図 11 に示す。これも前結果と同様に Host1 におけるスループットと受信したデータサイズの変化を示している。0-60 秒の間は Host2 では GET 要求が発生しない。このため資源競合は発生せず Host1 は Host2 の PHS 通信カードも同時に利用してデータ送受信を行うことができる。60 秒を経過した時点で Host2 で GET 要求が発生するため、この要求を受け取った時 Host2 上の SHP は Host1 から受信した要求に対するデータ受信を中断する。そして 150 秒経過すると Host2 では

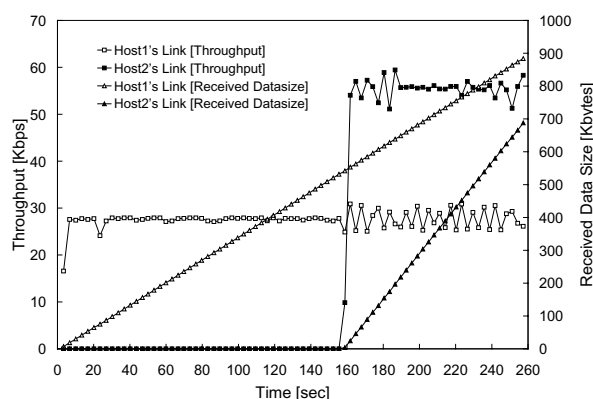


図 10 実験 3 (1) の結果

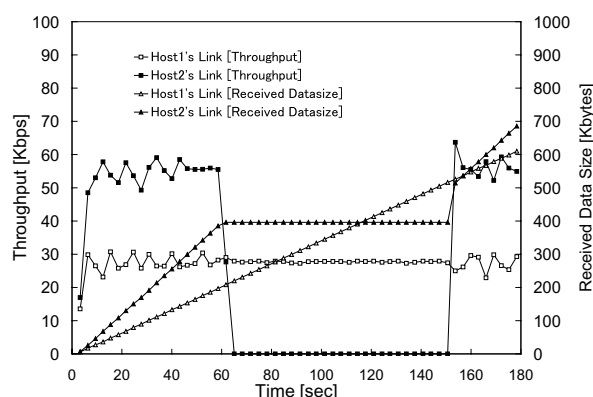


図 11 実験 3 (2) の結果

同端末上で発生した GET 要求に対するデータ送受信が完了し、再び Host1 に対するデータ送受信が再開されている。

本実験により同時要求や通信途中の要求発生など、複数の端末による共有資源の競合が発生した時でも、各端末が保持する NetI/F 資源はその端末で最大限利用することが可能であり、他の端末によって資源を占有され続けられないことが確認できた。

7. ま と め

ヘテロジニアスな環境下において、複数の端末がそれぞれ持つ複数の NetI/F 資源を共有するために必要となる共有資源の発見・管理機構：SCCM の実装を行った。そして SCCM を Web SHAKE に適用し、動的なクラスタの参加・脱退、通信途中の共有 NetI/F 資源の増減や品質の変動に対応できるようになったことを示した。さらに複数のユーザによる共有資源の同時利用時において、ある端末の資源が他の端末から一方的に利用されない仕組みを提案・実装し、提案方式が有効に動作していることを確認した。

SCCM では共有可能な NetI/F 資源を発見するために、各端末が定期的に資源情報をブロードキャストしている。

しかし常時資源情報を告知すれば電力が消費し、特に小型の端末の利用を想定した場合において課題が残る。また今回 SCCM で適用した Web SHAKE では、Web の 1 つ 1 つの GET 要求が、同一のサーバに対するものとは限らないという理由から End-to-End の遅延、帯域の測定結果をトラフィックの分配に用いていない。同一ホストと継続的な通信を行うような場合においては、End-to-End の遅延や帯域を考慮したトラフィック分配が必要であろう。

今後の課題として、バッテリー消費量を考慮した資源情報の告知手法、またクラスタ内の遅延、帯域だけではなく End-to-End での遅延、実効帯域を測定したトラフィック分配について検討する。

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 (1999)
- 2) 小西洋祐, 石原進, 水野忠則, "通信回線共有方式のためのクラスタ資源管理に関する検討," 情報処理学会研究報告, モバイルコンピューティングとワイヤレス通信研究会, 2002-MBL-23, Vol.2002, No.115, pp.172-178 (2002).
- 3) 小西洋祐, 石原進, 水野忠則, "通信回線共有方式を用いた高速 Web アクセスの実現," 情報処理学会論文誌, Vol.43, No.6, pp.1968-1976 (2002)
- 4) K. Sklower, B. Liyod, G. McGregor, D. Carr "The PPP Multilink Protocol (MP)," RFC1990 (1996)
- 5) Y. Kamio, F. kojima, M. fujise, "Implementation and Performance Evaluation of 384kbps-PHS Experimental system," IEICE, Vol. E83-B, No. 8, pp.1844-1853 (2000)
- 6) 林孝典, 山崎真一郎, 森田直人, 相田仁, 武市正人, 土居範久, "インターネットを用いた複数経路データ伝送方式の性能評価," 信学論 (B) Vol.J84-B, No.3, pp.523-533 (2001)
- 7) 日野哲志, 湧川隆次, 植原啓介, 村井純, "計算機群における" 動的なインターネット接続性" の共有に関する研究", 情報処理学会第 10 回マルチメディア通信と分散処理 (DPS) ワークショップ, pp. 69 - 74 (2002)