

通信回線共有方式の耐障害性の検討

飯田峰彦[†] 石原 進[‡] 井手口哲夫[§] 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[‡] 愛知県立大学[§]

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。しかし、複数の送信ホストが同時に回線を共有したときの適切なパケット振り分け方式についてはわかっていない。そこで、本論文ではシミュレーションによってこれを検証する。また、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、より効率的な障害回復方法について検討する。

Examination of the fault tolerance for Sharing Multiple Paths System

Minehiko Iida[†], Susumu Ishihara[‡], Tetsuo Ideguchi[§] 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. However the appropriate method to disperse packets is not clear. In this paper, we examine the problem by simulation. Furthermore we enumerated the problems of SHAKE prototype, examined more effective method for fault recovery.

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、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

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 [4, 5] 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 [6]。

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。この 2 つのネットワークの違い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

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

[‡] Faculty of Information, Shizuoka University

[§] Aichi Prefectural University

ネットワークへの接続回線を持っている点である。

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が、クラスタ内の送信ホストは1つであると仮定しており、複数の送信ホストが同時に回線を共有した場合の状況はまだわかっていない。また、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。しかしながら、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などによる通信を行った方がTCPのオーバーヘッドを減らすことができ、柔軟な誤り制御、再送制御が可能となる。

そこで本稿では、クラスタ内の複数のホストが共有回線を利用して通信する場合のケット振り分け方式、および効率的な信頼性保証方式について検討する。以下第2節では通信回線共有方式について述べ、第3節でケット振り分け方式について検討し、第4節で耐障害性の検討を行い、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(図1)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

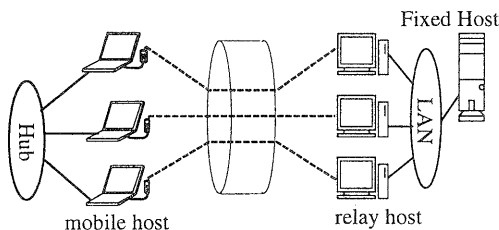


図1 通信回線共有方式の構成例

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して送信するのではなく、ある一定サイズの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、再整理)が従来の通信よりも増加する。

クラスタは理論上、どの通信媒体であっても形成可能であると考えている。クラスタ形成に使われる通信媒体としては次のものが考えられる。

- 有線 LAN (10~100Mbps)
転送レートは最も高いが、有線であるためネットワーク形成に汎用性がない。(ケーブルが必要)
- 無線 LAN (24GHz帯:2Mbps,19GHz帯:25Mbps)
無線であるためトポロジの制約がなく転送レートも高い。利用可能範囲が狭いがクラスタを利用する状況においては利用可能である。
- 携帯電話 (9.6Kbps)
現在最も一般に普及している通信媒体。しかしながら、転送レートが低いため、電話としては利用できるがデータ通信への利用は困難である。
- PHS (32K~64Kbps)
電話としても広く利用されているが、デジタル通信のため外出先からネットワークへ接続するためによく利用される。転送レートはモデム並みなので一般的な利用は可能である。ただし、アクセスポイントへ接続できる数に制約がある。
- IrDA (IrDA1.0: 2.4Kbps~115.2Kbps, IrDA1.1: 1.15Mbps~4Mbps)
赤外線による通信。現在のノートPCは赤外線ポートを標準で搭載しているので、ネットワーク形成のためのハードウェアを別途用意する必要がないという利点がある。転送レートは低いですが、狭い範囲内での利用ならばクラスタ形成は十分可能である [7]。

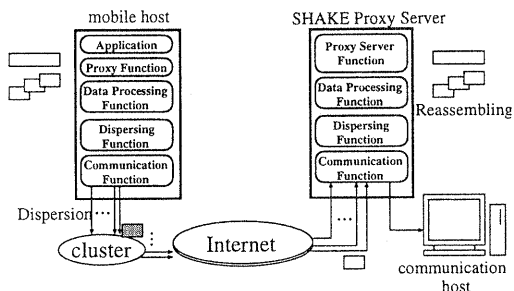


図2 SHAKEの機能構成

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。SHAKEの機能構成を図2に示す。SHAKEはクラスタを形成するすべてのモバイルホストと、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の4つの機能を持つ。それぞれの機能について以下で簡単に述べる。

(1) プロキシ通信機能

HTTP, FTPのクライアントアプリケーションがプロキシサーバとしてSHAKEを指定できるようにするためのものである。SHAKEをプロキシに設定することで任意の固定ホストはSHAKEを組み込むことなくSHAKEの機能を利用できる。また、従来のプロキシサーバのようにキャッシュ機能を持たせることで外部ネットワークへの負荷を軽減できる。

(2) 通信制御機能

クラスタおよび外部通信路のコネクション管理、経路情報の管理を行う。

SHAKEはクラスタ形成時にクラスタ内端末間のTCPコネクションをメッシュ状に設定する。クラスタ内から外部への経路情報を管理するため、SHAKEは各経路でのRTTを測定している。RTTは送信側の送信時のタイムスタンプ T_s と、受信側から送られてくる受信時のタイムスタンプ T_d を含んだレポートパケットによって T_d と T_s の差を算出することで求められる。[3]ではRTTの測定を行っているが振り分け制御などのパラメータとしての適用は未実装である。

(3) 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。再整列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。

(4) 振り分け機能

SHAKEでは外部通信路の通信媒体を自由に選択できることを前提としているので各端末ごとに転送速度、伝送遅延が異なる。従って、パケットを分散配信するときどのようにしてこれらを各通信路に振り分けるかが問題であり、クラスタ内および外部通信路の転送速度とも密接に関連している。また、パケットの振り分け法によって再整列に要する時間が変わってくる。

考えられる振り分け方式としては以下のものが挙げられる。なお、詳細については次節で検討する。

- ラウンドロビン方式による振り分け
- RTTによる振り分け
- 接続時の使用可能帯域を基準に振り分け

3 パケットの振り分け方式の検討

これまでは、本通信方式の実環境における評価は1台のモバイルホストからインターネット上の固定ホスト1台への通信で、振り分けはラウンドロビン方式で実験を行ってきた[3]。本方式は互いに他のモバイルホストの持つ回線も利用して通信する方式であるから、今後の実装にあたって複数ホストによる送信、固定ホストからモバイルホストへの通信についても検討する必要がある。

例えば、2つのモバイルホストが同時に送信を行う場合、モバイルホストが1つの場合に比べ、通信路を流れるデータが増加する。複数ホストによる送信の場

合、送信するホストが1台増えることによる他のファイル転送への影響は、制御パケットの場合よりもずっと大きくなる。したがって、それぞれのホストの送信時間は単一ホストで送信したときよりも送信時間が長くなることが考えられる。

そこで、本節では複数の送信ホストが同時に回線を共有しあうときのパケット振り分け手法について検討する。

3.1 経路の適切な選択

パケットを振り分ける時、送信ホストは利用する経路を選択する。現時点ではファイル転送には全ての外部通信路を用いることを想定しているが、[1]で検証したように、クラスタ内の転送速度と外部通信路の関係

$$T_{cluster} > \sum_{k=1, k \neq sender}^n T_k \quad (1)$$

($T_{cluster}$: クラスタ内転送速度, T_k : ホスト k が持つ外部通信路の転送速度) を満たさない、すなわちクラスタ内転送速度が外部通信路の転送速度の和よりも小さい場合、振り分けることでかえって送信時間が長くなってしまふ。このような場合、通信路を適切に選択して使用するクラスタ外への経路を少なくすることによって通信速度を向上できる。

3.2 ラウンドロビン方式による振り分け

各経路にパケットを等分配する方式。アルゴリズムは最も単純であるが、本通信方式ではスループットが最も転送速度が低い通信路に依存する。また、パケットの再整列のオーバーヘッドが大きくなることが予想される。また、転送速度が遅い経路は輻輳が頻繁に発生し、速い経路はパケットが流れていない時間が多くなってしまい、複数の回線全体としては帯域を有効利用できない。

3.3 重み付けによる振り分け

複数の外部通信路への経路それぞれに重みを与え、各通信路に振り分けるパケット長を調整する方式。振り分け方法としては次の2つの方法が考えられる。

- データを等間隔に分割して、重みによってパケットサイズを制御する方法 (図3)
- 分割されるパケットサイズは固定にして、重みによってパケット数を制御する方法 (図4)

重みによってパケットサイズを制御する場合、定期的にRTTの情報をクラスタメンバから受信してデータを送信しながら、振り分け率を変動させることができ、より厳密な振り分けが可能となる。しかしながら、パケットサイズには上、下限があるため、この方式で制御できるパケットサイズには制約がある。これに対し、パケットサイズを固定にする場合はパケットの分割は1段階で済み、制御も簡単である。

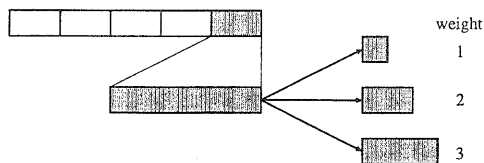


図3 パケットサイズを動的にする場合

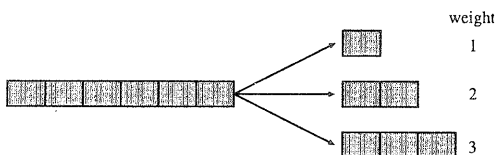


図4 パケットサイズを固定にする場合

RTTを重みにして振り分け

経路毎に測定されるRTTを重みにして振り分ける。ただし、RTTは常にランダムに変化するものでその度に各経路に振り分けるパケットサイズを計算するのが理想である。しかし、振り分けの計算がオーバーヘッドになる可能性があるので計算する間隔を調整する必要がある。通信路 i のRTTを R_i とすると、通信路 i へのパケット振り分け率 P_i は

$$P_i = \frac{1/R_i}{\sum_{i=1}^n 1/R_i} \quad (2)$$

で表される。

この振り分け方法であれば、途中でクラスタ内のホストの増減が起こっても柔軟に対応できる。

接続時の使用可能帯域を重みにして振り分け

接続時に分かる経路毎の使用可能帯域を利用して振り分ける方法。通信路 i の使用可能帯域を B_i とすると、通信路 i へのパケット振り分け率 Pw_i は

$$Pw_i = \frac{B_i}{\sum_{i=1}^n B_i} \quad (3)$$

で表される。

この振り分け方法は接続時の状態で振り分け率を決定している。接続後は振り分け率が一定であるため、ホストの増減により帯域が変化するような状況には対応できない。

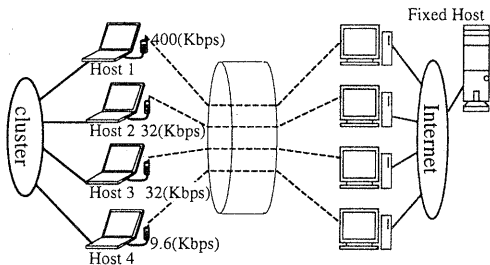


図 5 シミュレーションで想定した環境

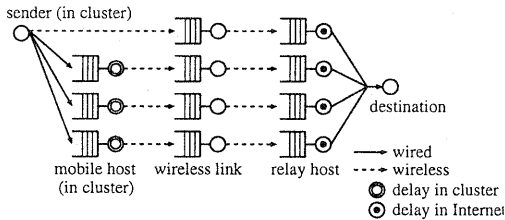


図 6 シミュレーションモデル

3.4 シミュレーションによる検証

振り分け方式としてラウンドロビン方式と RTT を重みとして振り分けた場合の性能をシミュレーションにより比較した。パケットは固定長で分割され、各経路に振り分けられるパケット数は重みを基準にする。シミュレーションは図 5 に示すように、クラスタ外部への通信速度がそれぞれ異なる 4 つのモバイルホストで一つのクラスタを構築した場合を想定した。端末 1~4 が持つ外部通信路の転送速度はそれぞれ、400(Kbps), 32(Kbps), 32(Kbps), 9.6(Kbps) であるとした。クラスタ内の転送速度は十分速いものとし、複数のホスト (4 台) が同じサイズ (750KByte) のデータを送信する。各ホストがデータ送信イベントをランダム (送信イベント間隔は平均 4 分の指数分布に従う) に発生させたときの各イベントにおけるファイル送信時間を測定した。シミュレーションモデルを図 6 に示す。なお、RTT は以下の計算式で得られるとした。

$$R_i = 0.76R_{i-1} + 0.24M \quad (4)$$

振り分けに用いる RTT はシミュレーションでは、パケットが宛先で受信されたら受信側がレポートパケットを送信元に送ることで測定しており、その RTT から計算された結果を RTT の推測値にしている。なお、 M は実測値、 R_{i-1} は R_i が得られる前の時点での RTT の予測値である。

結果と考察

ラウンドロビン方式で振り分けた場合のシミュレーション結果を図 7 に、重み付けで振り分けた場合を図

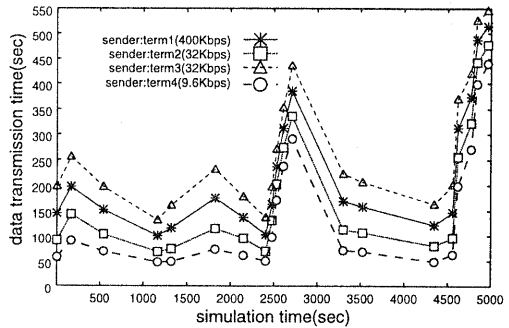


図 7 ラウンドロビン方式で振り分けた場合

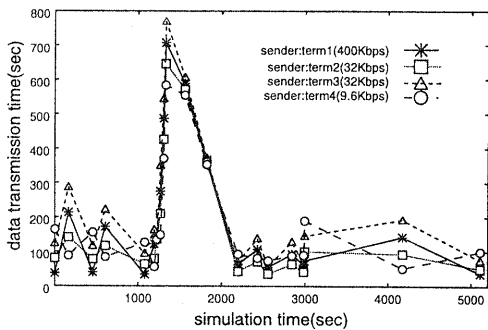


図 8 重みによって振り分けた場合

8 に示す。

ラウンドロビンで振り分けた場合、送信時間の順序関係が一定 (端末 4,1,2,3 の順) であった。そして、送信要求が頻繁に発生するとパケットが溢れ、時間を追うごとに送信時間が長くなっている。一方、RTT を重みとして振り分けた場合は送信イベント間隔が短くなくてもパケットが溢れることなく、各端末の送信時間もラウンドロビン方式に比べて短い値を維持できている。

また、表 1 に各ホストのファイル送信の平均時間を示す。送信時間の公平性という観点から見ると、それぞれの方式で送信時間が最長の端末と最短の端末の差はラウンドロビン方式で 128(sec)、重み付けで 58(sec) であり、RTT を用いて振り分けた方が送信時間がより公平になることが確かめられた。

表 1 各送信ホストの送信時間の平均値

	ラウンドロビン	重み付け
端末 1(400Kbps)	225(sec)	190(sec)
端末 2(32Kbps)	179	170
端末 3(32Kbps)	270	232
端末 4(9.6Kbps)	142	179

4 耐障害性の検討

これまで [1], および前節でシミュレーションによって通信回線共有方式の性能を理論的に評価してきた。しかし、これらの評価ではパケットロスを考慮していない。通信回線共有方式ではパケットは無線と有線の両方を通り、一般に無線部分では有線部分に比べてパケットロス率が高い。そこで、通信回線共有方式でパケットロスが発生したときの対処法について検討する。

4.1 プロトタイプの問題点

現時点での SHAKЕ 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 9)。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とをつなぐ 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 10)。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再送制御が困難である。

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 10) 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

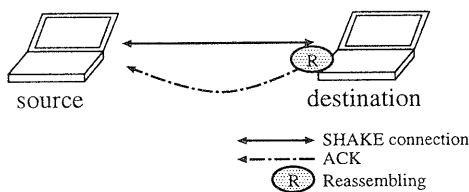


図 9 SHAKЕ コネクション

トの再整理が行われている。さらに、SHAKЕ 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 10 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、最後に宛先の SHAKЕ 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。

4.2 プロトコルの枠組

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。

プロトコルは UDP の上に新しいプロトコル (SHAKЕ) を設置する。SHAKЕ 層の低位層を UDP にすることで、パケットの振り分け、再整理の二重のオーバーヘッドをなくす。SHAKЕ 層は信頼性のある通信を実現する層 (SHAKЕ 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 (SHAKЕ B 層と呼ぶ) の 2 階層で構成される。すなわち、SHAKЕ A 層は TCP/IP における TCP、SHAKЕ B 層は IP に相当する機能を担当する。そして SHAKЕ 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

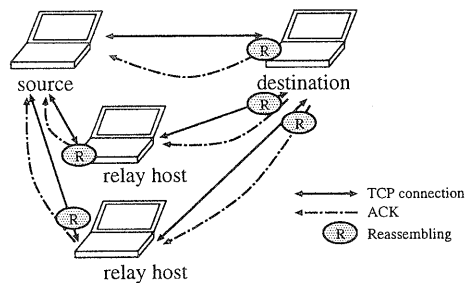


図 10 TCP コネクション (SHAKЕ プロトタイプ)

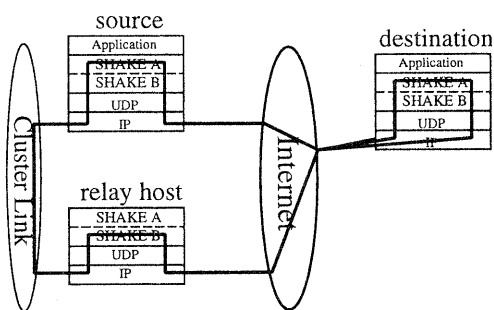


図 11 SHAKE プロトコル階層

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 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。

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、3 つの項目

- コネクション（経路）識別子
- クラスタ内のホストのポート番号または宛先のポート番号
- 経路の状態を示すパラメータ (RTT など)

を持つ。

例えば、クラスタ内のホスト A がインターネット上のホスト B にデータを送信する場合、セグメントを自分の持つ外部通信路に送るのか、クラスタを経由するのかを判断する。そのため、SHAKE B 層ではコネクションの識別子とポート番号の組および経路の状態によって経路を切替える。すなわち、第 3 節で検討した動的なパケット振り分けは SHAKE B 層が担当する。

パケット再整理

SHAKE A 層では図 9 のようにエンドツーエンドに一つのコネクションが存在すると見なす。したがって、これは TCP コネクションでの通信と同様であるから、パケットの再整理処理は TCP と同様になる。すなわち UDP および SHAKE B 層を通して受け取られたセグメントは、SHAKE A 層にてまとめられ、正しい順序で上位アプリケーションに渡される。

確認通知

複数経路を同時に使用する環境では、パケットが宛先に正しく届けられたことを送信元に通知する (ACK の送信) 方法として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

ACK によって送信元はパケットが正しく宛先に送られたことを知ることができると同時に、RTT の測定も可能である。しかし、TCP では受信したセグメントのシーケンス番号の最大値を ACK として送ることでまとめて確認通知が可能である。しかしながら、この方法では個々の経路に対し、シーケンス番号の管理をしなくてはならない。

- RTT が最短の経路に ACK を送信する方法

ACK の送信には RTT が最短の経路を利用する。経路毎の制御ではないので経路数が n のとき、 n 個の経路をまとめて扱う。すなわち、TCP と同じく、SHAKE A 層が確実に受け取ったデータ列に対し、ACK を SHAKE B 層を介して配送し、SHAKE B 層が適切な経路を選択する。ただし、ACK を RTT の測定には使えないので、別途 RTT を測定するための制御パケットを流す必要がある。

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の受信側の SHAKE A 層で分かっている必要がある。しかし、再送制御に経路制御を組み合わせるのには 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。

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。

一方で、この方法を用いた場合、クラスタ内の中継ホストにキャッシュされたパケットを再送に用いることも可能である。しかしながら、再送は経路

毎に行わなければならないがあり、再送を促す、再送タイムアウト、再送要求の送信などは各経路毎に制御する必要がある。また、クラスタ内の転送速度は外部通信路に比べてずっと速いと仮定しているのでもキャッシングによる効果はさほど期待できない。

- 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、受信側の再整理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

$$RTT = \alpha RTT + (1 - \alpha)M \quad (5)$$

$$D = 7/8D + 1/8|RTT - M| \quad (6)$$

$$Timeout = RTT + 4D \quad (7)$$

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 層で RTT を測定する場合、ACK がどの経路を通過してきたのかを SHAKE A 層は知らない。もし ACK が最速の経路を通るとすると、転送速度が最低の経路を通ったパケットに対する RTT は、SHAKE A 層では実際の RTT より短くなり、正しい RTT の測定ができない。したがって、RTT の測定および再送タイムアウトの算出は SHAKE B 層で行い、結果を SHAKE A 層が受け取るのが望ましいと考えられる。SHAKE B 層で再送タイムアウトを決定する方法としては、 n を経路数とすると、

- RTT が最長の経路の実測 RTT に $1/n$ を掛けたものを式 (5) の RTT 予測値とする
- 各経路の実測 RTT の平均値に $1/n$ を掛けたものを式 (5) の RTT 予測値とする

などが考えられる。

再送制御は再送タイムアウトを基準に行われるが、SHAKE A 層で算出するのは困難である。そこで再送タイムアウトの算出だけは、RTT を経路毎に測定し

ている SHAKE B 層で行う方が得策である。しかし唯一の欠点は、プロトコルの階層化理念に反することである。

SHAKE B 層で再送タイムアウトを決定する方法を 2 つ挙げたが、理論的解析および検討は今後の課題である。

5 おわりに

本論文では、まず複数ホストの通信時間の公平性の検証を行った。その結果、RTT を重みとして振り分けることで転送時間を短縮でき、送信ホスト間の転送時間の差を小さくできることがわかった。

また、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上で対障害性について検討した。今後、振り分け方式、再送方式についてより詳細な検討をシミュレーションを用いて行う予定である。

参考文献

- [1] 飯田峰彦, 峰野博史, 太田賢, 井手口哲夫, 水野忠則, “通信回線共有方式におけるクラスタ内転送速度の検討”, マルチメディア; 分散; 協調とモバイル (DICOMO'99) シンポジウム, Vol.99, No.7, pp.231-236.
- [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] Vincent D. Park and M. Scott Corson. “A highly adaptive distributed routing algorithm for mobile wireless network.” In *Proceedings of INFOCOM'97*, April 1997
- [5] 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)
- [6] 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.
- [7] 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.