

# モバイル通信を指向した Dynamic Multi Link TCP の設計

殿内雅晴<sup>†</sup> 鄭宇新<sup>†</sup> 峰野博史<sup>††</sup> 石原進<sup>†††</sup> 高橋修<sup>††††</sup> 水野忠則<sup>††</sup>

<sup>†</sup> 静岡大学大学院情報学研究科 <sup>††</sup> 静岡大学情報学部 <sup>†††</sup> 静岡大学工学部 <sup>††††</sup> NTT ドコモ

## Design of Dynamic Multi Link TCP for Mobile Communication

Masaharu Tonouchi<sup>†</sup> Ushin Tei<sup>†</sup> Hiroshi Mineno<sup>††</sup> Susumu Ishihara<sup>†††</sup>  
Osamu Takahashi<sup>††††</sup> Tadanori Mizuno<sup>††</sup>

<sup>†</sup> Graduate School of Information, Shizuoka University <sup>††</sup> Faculty of Information, Shizuoka University  
<sup>†††</sup> Faculty of Engineering, Shizuoka University <sup>††††</sup> NTT docomo

### 1 はじめに

近年、無線 LAN ホットスポットや PHS 網、携帯電話網などの無線通信インフラが整備されるだけでなく、携帯端末にも有線 LAN、無線 LAN、Bluetooth、IrDA といった複数のネットワークインタフェースが装備されることが多くなった。ユーザはこれら複数のネットワークインタフェースを状況に応じて使い分けて利用することができるが、同時に複数のネットワークインタフェースを用いて通信の高速化、高信頼化を図ることもできる。また、通信回線共有方式 SHAKE (SHAring multiple paths procedure for cluster network Environment) [1] のように携帯電話などの長距離無線リンクを持つ端末が集まり、各端末が持つ長距離無線リンクを共有することで通信の高速高信頼化を図ることもできる。

本稿では、このように性能や品質の異なる複数の経路を利用し通信の高速高信頼化を目指す通信方式において、トランスポート層で複数経路通信に適した輻輳制御や再送制御を行う Dynamic Multi Link TCP (以下 DMTCP とする) を提案し、その機能の詳細な設計を行う。

### 2 DMTCP

#### 2.1 複数経路を用いた通信方式

複数の経路を用いた通信方式として、インターネットを用いた複数経路データ伝送方式 [2] がある。これはインターネットにおいてネットワーク障害に対する信頼性やスループットの向上を目的とし、専用ゲートウェイを用いて複数経路に IP トンネリングを行うことで実現されている。この方法では特別に専用ゲートウェイを設置する必要があり、またトランスポート層での送信制御までは考慮されていない。MulTCP [3] では大容量のデータ転送を行う際に、輻輳ウィンドウサイズの極端な変化を抑え安定した送信を行うために、一つのコネクションの中に複数のサブコネクションを用いる通信に適した送信制御を提案している。しかしこの方法は複数経路の通信を想定したのではなく、通信中の経路の増減や、各サブコネクションの経路品質の差を考慮に入れた送信制御などは検討されていない。

一方無線による複数経路を束ねて通信する方式として SHAKE がある。SHAKE では、複数の移動端末が短距離高速リンクを用

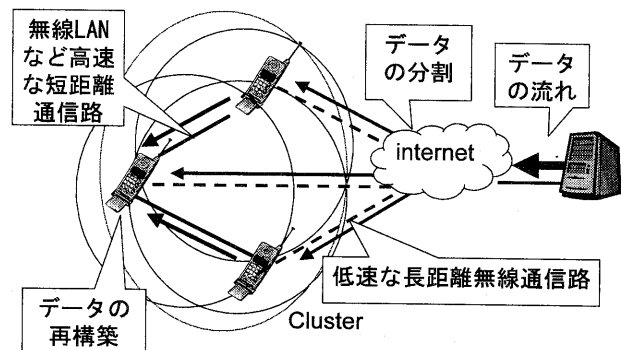


図 1: 通信回線共有方式

いて相互に接続し、一時的にネットワークを構成することで (これをクラスタと呼ぶ)、ある端末がクラスタ外のホストと通信する時は、それ自身が持つ通信路のみならずクラスタ内の他の端末の外部への通信路も用いて通信を行う方式である。無線を利用した通信は、その特性上、有線通信に比べて帯域が狭く、また伝送誤りが生じる割合が高いため、バースト的なデータロス、スループットの低下が起こるといった課題も存在する。SHAKE では、通信の高速化を実現するだけでなく、品質の変化しやすい無線通信においても代替経路を利用することで信頼性を向上させることができる (図 1)。

現在様々な階層からのアプローチでこの SHAKE の研究が進められているが、トランスポート層において TCP の機能を拡張し、この SHAKE に適したものに拡張したプロトコルとして筆者らは TCP SHAKE [4] を提案している。TCP では通信の信頼性の保証と、再送制御や輻輳制御などの送信制御が行われている。このようにトランスポート層において複数経路通信を制御することで柔軟で動的なトラフィック分配や再送制御が可能となる。TCP SHAKE ではトランスポート層において SHAKE に適した送信制御を行うことで効率的な通信手法を提案しているが、想定されている利用方法が SHAKE の送信制御に限られており、また通信途中に経路数が増減することについては考慮されていない。

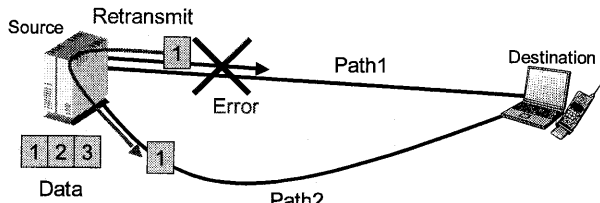


図 2: DMTCP の再送制御

## 2.2 DMTCP

DMTCP では、複数の経路を明示的に使い分けることのできるよう拡張されたネットワーク層を前提としている。1 台の端末が複数のネットワークインタフェースで異なる経路を使う通信方式であれば、経路とネットワークインタフェースの対応付けを行う機能が必要であり、SHAKE のようにクラスタ内の他の端末の通信路を利用する通信方式であれば、経路と中継ホストの対応付けを行う機能が必要となる。本稿ではそのような機能が既にあることを前提としている。

DMTCP ではエンドエンド間の複数経路を一つのコネクションとみなし、経路が複数あることを活かして通信の効率を向上させるために以下のような送信制御を行う。

**経路ごとの輻輳制御によるトラフィック分配** TCP における輻輳ウィンドウとは、通信経路の輻輳状態に応じて送信可能なデータ量を調整する機構である。通信経路が複数ある場合に輻輳ウィンドウが一つでは、帯域やロス率など各経路の品質に応じた効率的な輻輳制御を行うことはできない。例えば一つの経路で輻輳が起きた場合には全ての経路で送信量を落とすことになる。DMTCP では経路ごとに輻輳ウィンドウを設置することにより、この輻輳制御を経路ごとに行う。これによって送信エラーの起きた経路では輻輳ウィンドウが縮小し送信量を減らすことができ、順調に確認応答が返ってくる経路では輻輳ウィンドウが増加し送信量を増やすことができる。これは送信セグメントを性質の異なる経路に適切に振り分けることになる。DMTCP では輻輳制御の仕組みを利用して各経路の状況に応じた動的なトラフィック分配を実現する。

**複数経路を活用した再送制御** 複数の経路を利用した通信では、送信エラー発生時に、送信エラーの発生した経路で再送を行う必要はない。つまり、再送時により適切な経路を使って再送することができる。例えば、送信エラーが輻輳によるものであれば、その経路で再送を行っても再びそのセグメントが失われてしまう可能性が非常に高い。別の経路を使って再送することによって再送の信頼性を向上させることができる。DMTCP のようにエンドエンド間で一つのコネクションを形成していれば、複数経路であることを活かして元々送信した以外の経路で再送を行うことができる。この様子を 図 2 に示す。この図では経路 1 で送信したセグメント 1 が喪失している。この時経路 1 は再送が成功するように経路 2 でこのセグメントを再送している。

**通信中の経路の増減への対応** 複数経路を用いた通信では、通信の最中にある経路が一時的に使用不能な状態になるなど利

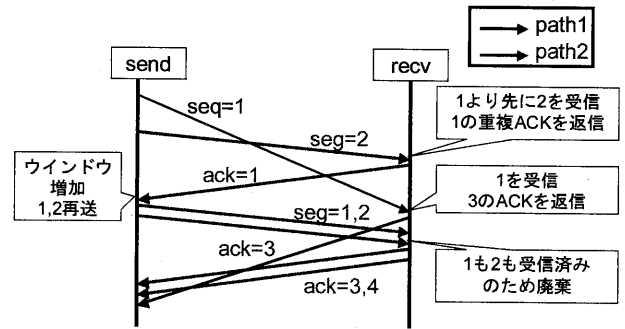


図 3: 複数経路でシーケンス番号一つの場合

用可能な経路数が増減することがある。特に SHAKE などのモバイル通信の場合ネットワーク構成が頻繁に変化することも考えられ、使用可能な経路数を固定したものとして考えると、経路が切断した場合に通信の信頼性を保証できないという問題が生じる。DMTCP ではコネクション確立時のタイミングで使用可能な経路の増加に対応するとともに動的な経路の減少に対応している。

## 3 DMTCP の設計

### 3.1 確認応答制御

DMTCP のように帯域や遅延など品質の異なる複数の経路を用いて通信を行う場合、単一経路での通信に比べセグメントが不連続に受信される可能性が高くなる。受信側でセグメントが断続的に受信されると、送信側は重複 ACK を頻繁に受信することになる。送信側はセグメントの喪失を検知すると喪失したセグメントの再送を行うが、送信セグメントのうち、どのセグメントが受信され、どのセグメントが喪失したのかを送信側は詳細に知ることができない。したがって、受信側が既に受信しているセグメントを複数再送する可能性があり、重複データの送信に無駄が生じる (図 3)。

DMTCP では送信セグメントに対し、経路ごとにシーケンス番号を付けることでこの問題を解決している。これをローカルシーケンス番号と呼ぶ。また DMTCP は TCP で用いられる通常のシーケンス番号も併用している。ローカルシーケンス番号と区別するために通常の TCP で用いられるシーケンス番号をグローバルシーケンス番号と呼ぶこととする。このローカルシーケンス番号を用いることで、各経路でセグメントが順序通り受信されているか、どのセグメントが喪失したかを把握することが可能となる。複数経路を用いた通信では、各経路で遅延やパケットロス率が異なるため、受信側ホストに到着したセグメントの順序の誤りが頻発するが、各経路では通常の TCP と同程度しか発生しないため、ローカルシーケンス番号によって経路ごとの順序誤りを抑制することができる。

再送制御をグローバルシーケンス番号だけを用いて行くと、喪失したセグメントのうち最小のグローバルシーケンス番号を持つセグメントしか重複 ACK は発生しないことになり、高速再送アルゴリズムによって再送を行うのはそのセグメントのみとなる。複数の経路で同時にセグメントが喪失した場合、高速

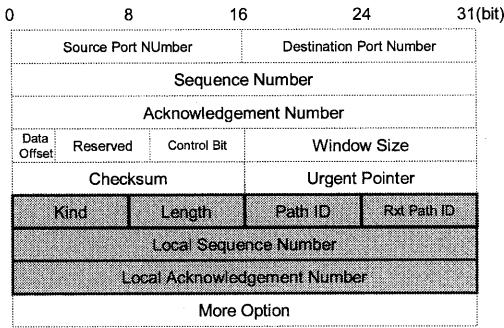


図 4: DMTCP ヘッドオプション

再送アルゴリズムによる再送は一つの経路でのみ行われ、他の経路ではセグメントのタイムアウトを待つこととなり通信の効率が悪化する。ローカルシーケンス番号を用いると、各経路で重複 ACK が発生することになり高速再送アルゴリズムを全ての経路で実行することができる。

DMTCP はローカルシーケンス番号を管理するために TCP ヘッドのオプションフィールドを使用して、経路 ID(Path ID)、再送経路 ID(Rxt Path ID)、ローカルシーケンス番号 (Local Sequence Number)、ローカル ACK(Local Acknowledge Number) を DMTCP オプションとして設定する。この様子を図 4 に示す。経路 ID は送信セグメントがどの経路で送信され、どの経路の送信ウィンドウにより管理されているかを示す。再送経路 ID は、受信側が再送セグメントを実際にはどの経路で受信したか知るためのものである。前述したように DMTCP では最初に使用した経路とは異なる経路で再送することが可能であるが、その場合この再送経路 ID で再送時に実際に使用した経路を知ることができる。セグメントを実際に受信した経路を使って ACK を返すことで、送信側は各経路の RTT を測定することができる。また送信側は実際に使用した経路で ACK を受信することにより、その経路が通信可能であることを認識できる。

### 3.2 再送制御

2.2 で述べたように、再送時に用いる経路は最初の経路とは限らず異なる経路を用いることができる。また、再送の経路を選ぶ方法は複数考えられる。

- 初期条件が最もよい経路

通信前にあらかじめ経路の特性から最も品質の高いと思われる経路でのみ再送する。経路の特性としては、帯域、遅延、ロス率、遅延のゆらぎなどが考えられる。しかし、通信中経路の状況は動的に変化することも考えられる。すなわち、最も品質の高い経路がダウンしてしまう場合や、この経路でのみ再送を行うため再送セグメントが集中してしまいそれによって輻輳が起きることが考えられる。

- RTT 最小の経路

DMTCP は各経路で RTT を測定しているため、この RTT の最小の経路を利用して再送を行う。この方法は経路の状況によって柔軟に経路を変更できる。しかし RTT が最小であっても帯域の小さな経路である場合もあり、セグメントサイズによっては必ずしも最適な経路になるとは限らない。

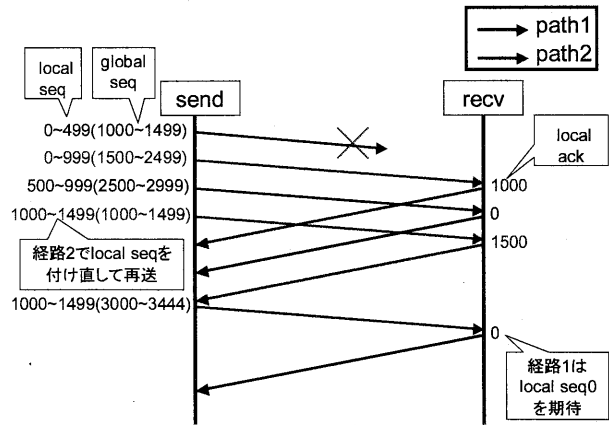


図 5: 再送時ローカルシーケンス番号を付け直した場合

- 最も輻輳ウィンドウサイズの大きい経路

DMTCP では各経路で輻輳制御を行っているため、通信の安定している経路ほど輻輳ウィンドウサイズは大きくなる。そこで最も輻輳ウィンドウサイズの大きな経路を最も通信の安定している経路と判断しその経路で再送を行う。ウィンドウサイズは経路の状況とセグメントサイズに応じて変化するため、上の 2 方式より適した経路を選択することができると考えられる。

再送時に最初に使用した経路とは異なる経路で送信した場合、再送セグメントのローカルシーケンス番号を再送経路で新たに付け直してしまうと問題が起こる。受信ホストは最初の経路上の欠けたローカルシーケンス番号を持ったセグメントを待ち続けてしまう。この様子を図 5 に示す。この例では送信側が経路 1 でローカルシーケンス番号 0~499 として送信したセグメントを経路 2 でローカルシーケンス番号 1000~1499 として再送している。この場合受信側は経路 1 でローカルシーケンス番号 0~499 を待ち続けてしまう。

DMTCP では再送セグメントには最初に使用した経路と同じ経路 ID とローカルシーケンス番号を付けるために、送信ホストは送信セグメントのローカルシーケンス番号とグローバルシーケンス番号の対応付けをとって再送に備える。これにより再送時には初回の送信時と同じローカルシーケンス番号とグローバルシーケンス番号の組み合わせでセグメントを送信することができる。DMTCP では、この対応付けを送信セグメントリストとして経路ごとに保存しておく。送信セグメントリストは、送信済みセグメントブロックのグローバルシーケンス番号の先頭とそれに対応するローカルシーケンス番号の先頭、このブロックのデータ長をメンバとして持つ二重連結リストとして構成される。送信済みセグメントのブロックとは、連続したグローバルシーケンス番号を持った送信済みセグメントの集まりのことである。送信セグメントリストの例を図 6 に示す。図 6 の例では経路 1 からグローバルシーケンス番号 1000~1499 と 2500~2999 という 2 つのセグメントブロックを送信し、経路 2 から 1500~2499 のセグメントブロックを送信しているということを示している。送信セグメントリストは確認応答されるまで送信側によって保持され、確認応答されたセグメントは送信セグメントリストから削除される。

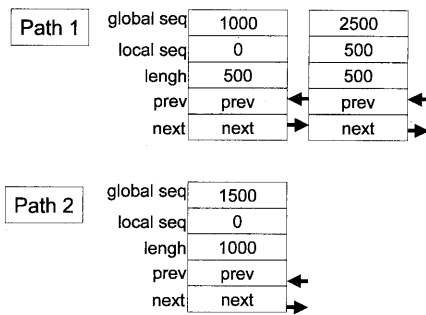


図 6: 送信セグメントリストの例

セグメントを受信したホストは、各経路ごとにローカル ACK を返す。経路で順序通りにセグメントが到着している場合には単純に順序通り届いた地点までのローカル ACK を返せばよい。しかし誤った順序で到着したセグメントがある場合はどのセグメントが欠けているかを把握し、順序通りに届いた地点までのローカル ACK を返さなければならない。そのために受信側では、経路ごとに誤った順序で受信したセグメントの一覧を受信セグメントリストとして保存する。受信セグメントリストは、受信セグメントのローカルシーケンス番号の先頭とこのブロックのデータ長をメンバとして持つ二重連結リストとして構成される。受信セグメントリストの例を図 7 に示す。誤った順序で届いたセグメントの再整列や受信ウィンドウサイズの管理はコネクション単位で、通常の TCP の機能によって行えるため、各経路はローカル ACK を返すだけでよい。この例では、経路 1 でローカルシーケンス番号 0~499 のセグメントを飛ばして 500~999 のセグメントが先に到着していることを示している。この時受信側のホストはローカルシーケンス番号 0 の重複ローカル ACK を返すことになる。その後 0~499 のセグメントを受信すると、ローカルシーケンス番号 1000 のローカル ACK を返し、受信セグメントリストの 1000 以下を解放する。

この二つのリストと輻輳ウィンドウ制御変数、タイマを制御する変数は、経路ごとに PCB(Path Control Block) と呼ばれるデータ領域を用意し管理する。

### 3.3 コネクション管理と経路の増減への対応

通常の TCP のコネクション確立手順では、各経路に必要なデータ領域の確保や、使用可能な経路の信頼性を保証することはできない。DMTCP ではコネクション確立時に通常の TCP の処理に加え、以下の処理を行う。

- PCB の準備

両端のホストは、各経路を制御する変数を管理するために経路の数だけ PCB を用意する。そのため両端のホストは、使用可能な経路数をネットワーク層から取得し、その数だけ PCB を用意する。

- 全ての経路を用いた 3 ウェイハンドシェイク

送信側は送信開始時に全ての経路を使って SYN セグメントを送信する。これを受け取ったホストは SYN+ACK セグメントを全ての経路から返す。送信側は各経路の PCB にコネクション確立タイマを用意しており、このタイマが

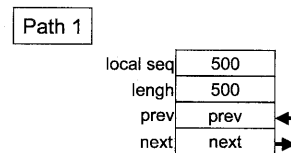


図 7: 受信セグメントリストの例

切れる前に ACK がなければその経路は使えないと判断し、PCB を削除する。これを行うことで、送信側はネットワーク層から提供された使用可能経路が実際に通信可能であるかどうかを確認することができる。

DMTCP は使用可能な経路の確認をコネクション確立時に行うため、そのタイミングで経路の増加に対応できる。また通信中に経路の一部が使用不能になった場合は、一つの経路におけるタイムアウトの連続発生回数が、与えられた閾値を超えた時にこれを検知する。この時使用不能になった経路で送られ、まだ確認応答の得られていない送信済みセグメントを全て他の経路で再送し、全ての再送が成功してから PCB を解放することで経路の動的な減少に対応する。

コネクションの切断は、TCP 同様ホスト間でお互いに FIN セグメントとそれに対する ACK を送信することで完了する。これらのセグメントは、通常データセグメント同様にホスト間で一つずつ送信し、複数の経路への送信は行わない。

## 4 まとめ

移動無線通信環境における複数経路を用いた通信において、性能や品質の異なる複数の経路を一つのコネクションとみなし適切な送信制御を行うトランスポート層プロトコル DMTCP を提案し、その詳細な設計を行った。DMTCP によって性能や品質の異なる複数経路通信に適した輻輳制御や再送制御を行うことができると考えられる。今後、本稿で検討した項目についてシミュレーションで評価し、本提案手法の有効性を示す。

## 参考文献

- [1] 峰野, 青野, 太田, 井手口, 水野 “クラスタ型ネットワークにおける通信回線共有方式の提案と評価” 情報処理学会論文誌, Vol.41, No.2, pp.354-362, (2000)
- [2] 林, 山崎, 森田, 相田, 武市, 土居 “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” 電子情報通信学会論文誌, VolJ84-B No.3, (2001)
- [3] P.Gevros, et al., “Analysis of a Method for Differential TCP Service,” available at <http://www.mice.cs.ucl.ac.uk/multimedia/publications/gi99.ps.gz>, December A1999.
- [4] 飯田, 石原, 水野 “複数無線リンク上でのコネクション型通信手法の性能評価” 情報処理学会研究報告 2001-DPS-102, Vol.2001, No.29, マルチメディア通信と分散処理 102-2, pp.127-132, (2001)