NAPTを越えた端末の移動時のTCPコネクション維持による移動透過性保証プロトコル

清水 智行

平成14年2月5日

概 要

本研究では、インターネットにおいてNAPT配下とグローバル空間との間でNAPTをまたいだ端末移動が行なわれる場合でも、TCPコネクションを維持することによって移動透過性を保証するプロトコルを提案する。

現在のインターネットにおいて、一つのIPアドレスを複数の端末が共有する、ないしファイヤウォールとして利用する目的で、NAPTがよく利用される。しかし、NAPTは端末のIPアドレスとポート番号を変換する機能とともに、外部からの接続要求を遮断する性質を持つ。このような状況ではネットワーク層による移動透過性の保証には限界があり、トランスポート層における保証手法が必要となる。

本研究では、NAPTの関与した移動時にも対処可能な再接続手順の制御を行なうための手法として、TCPコネクション維持プロトコルを提案する。このプロトコルは、端末の移動情報を交換するプロトコルと拡張されたTCP、さらにTCPコネクションを中継するプロキシで構成されている。これによって、NAPTの存在に関わらず、端末の移動後に通信を再開させることが可能となる。

1 はじめに

近年、インターネットは非常に広範囲で利用されるようになってきている。企業や研究機関のみならず、一般の家庭においても、電子メールの交換やWebの閲覧などの目的で日常的に利用されるようになりつつある。機器の小型化や低価格化、携帯電話や無線LANなどの普及に伴い、通信端末を持ち運んでさまざまな場所からインターネットに接続するという利用形態も一般的となりつつある。

しかし、インターネットに接続されている通信端末を移動させるとき、移動中に通信が途切れないようにすることは現状では困難である。例えば、イーサネットのケーブルでインターネットに接続された端末を移動させる時には無線LANに切り替え、外出するときには携帯電話によってインターネットに接続する、といったように、場所によって通信媒体を切り替えることが想定される。また、同じ建物内での移動においても、場所によってセグメントが切り替わったり、無線LANのアクセスポイントが切り替わったりすることによって、別々のDHCPサーバからIPアドレスを取得しなおすような場合も想定される。従って、通信端末の移動によって通信が途切れないように維持するには、IPアドレスが変化しても通信を維持できる仕組みが必要となる。

端末が移動したときに通信を維持する手法に関する研究は、これまでにも盛んに行なわれている。これらの手法は大きく2つに分けられる。一つは移動端末に固定IPアドレスを与える、ネットワーク層における解決手法、もう一つは移動によってIPアドレスが変化したときに、その変化後のIPアドレスにおいて通信を再開する、トランスポート層における手法である。

前者の手法として、例えば Mobile IP[1] がある。Mobile IP では、固定 IP アドレスを持った移動端末である Mobile Node(MN) は、Home Agent(HA) を経由して Corresponding Node(CN) と通信する。

MN が別のセグメントに移動したとき、移動先に Foreign Agent(FA) があれば、FA の IP アドレス (CoA; Care of Address) を HA に通知することによって、CN と MN が送受信するパケットを FA と HA の間でカプセル化して中継することができる。これによって、移動先においても MN は同じ IP アドレスを利用して通信を続けることが可能となる (FA モード)。

MN の移動先に FA が存在しない場合、DHCP 等によって獲得した IP アドレスを CoA として HA に通知し、MN 自身が FA として機能してカプセル化通信を行なうことが可能となる (Co-located Care-of Address モード)。

しかし、Mobile IP を利用するためには、MN が固定 IP アドレスを与え、HA を必ず設置しなければならない。従って、固定 IP アドレスを用意できる環境においては活用できるが、家庭のように固定 IP アドレスを持たない環境で利用することはできない。

また、MN の移動先に NAPT[4] が存在する場合、FA モードにおいて FA が NAPT の内側に存在するとき、もしくは FA が存在せず Co-located Care-of-Address モードを利用するとき、CoA として MN が HA に送信される IP アドレスは HA にとって意味のないものとなる。また、Mobile IP におけるカプセル化通信は上位層 (トランスポート層) が使用するポート番号等の情報を含まないため、NAPT はパケットの受け渡し先の端末を判別することができなくなる。従って、Mobile IP を利用するとき NAPT は利用できない。

これに対して後者の手法として、例えば Mobile TCP Socket[2] や An End-to-End Approach to Host Mobility[3] がある。これらは、移動先の IP アドレスに対して通信を維持するために、移動端末が相手端末に対して移動先の IP アドレスを通知し、移動先において移動直前の状態から通信を再開する、というものである。特に、TCP コネクションにおいては、TCP 自身による再送制御によって、移動後の通信再開が容易なものとなる。この手法には固定 IP アドレスが不要であり、端末間の経路上の機器は従来から利用されている機器をそのまま利用することができる。また、TCP コネクションの確立手順自体は従来の TCP と変わらないため、この手法を実装していない従来の端末との互換性も確保されている。

しかし、[2]、[3] のうち、いずれの手法においても、NAPT(Network Address Port Translator)[1][4] を越えた移動時に TCP コネクションを維持することができない。また、2 つの端末の移動する時間帯が重なった場合において、端末間の TCP コネクションを維持することも不可能である。

移動端末同士が通信する場合、互いに通信中の 2 つの端末が同じ時間帯に移動する場合も想定される。しかし、2 つの端末の移動する時間帯が重なった場合、両端末の IP アドレスが移動中に変化すると、互いに移動先の IP アドレスを通知することができないため、移動後に TCP コネクションを再開することが不可能となる。

また、家庭においてインターネットを利用する場合、インターネットサービスプロバイダからは一つだけしか IP アドレスの割り当てを受けられない場合が多い。このような環境で複数の端末を同時にネットワークに接続するために、NAPT を利用して一つの IP アドレスを複数の端末が共有するといった利用形態が一般的となりつつある。NAPT は、外部ネットワークからの接続要求を遮断する性質を持つため、企業などにおいてもファイヤウォールとして利用されることがある。

しかし、この性質は、言い換えれば NAPT 内に移動した端末に対して、外から TCP コネクションの再開要求を送信することが不可能であるということを意味する。従って、NAPT 内の端末が相手端末に対して再開要求を送信するように、通信の再開手順を制御する仕組みが必要となる。特

[1] 一般には IP マスカレードとも呼ばれる。

に、2 つの移動端末が互いに別々の NAPT 内に移動した場合は、互いに通信再開要求を直接相手に送信することが不可能となるため、両端末間の通信を中継する仕組みが必要となる。

また、複数の端末の IP アドレスが NAPT によって同じ IP アドレスに変換されるため、移動後に通信を再開する際、端末間の対応関係を維持するための仕組みも必要となる。

ポート変換に対処するためには、ネットワーク層ではなくトランスポート層における解決手法を設計する必要がある。そこで本研究では、2 つの端末の移動する時間帯が重なった場合と NAPT を越えた端末移動が発生した場合でも利用可能な、TCP コネクションの維持を支援するためのプロトコルを提案する。

2 つの端末の移動する時間帯が重なった場合に対しては、端末の移動先情報を交換するためのサーバを設置し、利用することによって、通信相手の端末が移動してもその移動先を知り、通信を再開することを可能とする。この 2 端末間ならびに端末-サーバ間の移動先通知の仕組みを移動先情報交換プロトコルと呼ぶこととする。また、このプロトコルでは、移動端末から受け取ったパケットと実際に通知された IP アドレスの比較によって、NAPT の存在の検出も行なう。

相手端末の移動先をサーバから取得したときは、移動した相手端末に対して TCP コネクションの再開要求を送信することで通信が維持される。　2 つの移動端末が互いに別々の NAPT 内に移動した場合は、直接通信を再開することはできないが、サーバがその状況を検出し、移動端末にプロキシと通信するように指示することによって、プロキシが TCP コネクションを中継して擬似的に TCP コネクションを維持することが可能となる。

以上の手法を組み合わせることによって、サーバが NAPT の有無を検出し、再開要求の送信手順を適切に判断して、TCP コネクション維持のために移動端末に適切な指示をすることが可能となる。

また、この方式の場合、移動端末にこのプロトコルを実装し、最低一つのサーバを設置することによって利用可能となるため、NAPT による IP アドレスとポート番号の変換に対処できるだけでなく、Mobile IP などに比べて、固定的な HA が不要となるため、必要となる機器の数が少なくなり普及時の手間が少ないという利点もある。

以下、第 2 章では端末移動時における移動透過性実現のための従来手法とその問題点について述べ、第 3 章で本研究が提案する端末移動時の TCP コネクションの維持手法について説明し、第 4 章で本手法を実装して実験した結果を示し、第 5 章で本研究の提案手法と実験結果をまとめて結論とする。

2　移動透過性の実現手法

一般に、通信端末を持ち運んだり、状況に応じて通信媒体を切り替えたりすることによって、端末が利用する IP アドレスが変化すると、それまでに行なっていた通信は切断される。しかし、ユーザが利用するアプリケーションにとっては、IP アドレスが変化しても通信を再開させたい状況も多い。

端末移動時に IP アドレスの変動を吸収して通信が切断されないように維持する手法は、これまでにも数多く提案されている [1][2][3]。それらは主に次に述べる 2 つに分類することができる。一つはネットワーク層における手法、もう一つはトランスポート層における手法である。本章では、これらの手法の概要を説明し、それらによって解決できない問題点を述べる。

なお、本論文において「端末の移動」とは、通信媒体の切り替えや、DHCP などのアドレス割り当ての仕組みによって、端末の使用する IP アドレスが変動することを指すものとする。

2.1 ネットワーク層における移動透過性

ネットワーク層における解決手法として代表的なものに、Mobile IP[1] がある。まず、Mobile IP の概要を以下に示す。

Mobile IP において、移動端末 (MN; Mobile Node) は Home Address と呼ばれる固定 IP アドレスを持つ。MN は通常ホームネットワーク[2]内で Home Agent(HA) を経由して通信する。MN がホームネットワーク外に移動するときは、移動先の Foreign Agent(FA) を通じて HA に移動先の FA の IP アドレス (CoA; Care-of Address) を通知する。

移動先の通知を受けた HA は、ホームネットワーク内の端末 (Correspondent Node) が MN と通信するときに、CN が MN に送信するパケットをカプセル化して一旦 FA まで送信し、カプセル化されたパケットを受信した FA は、MN に対してデカプセル化されたパケットを転送する (FA モード、図 1)[8]。

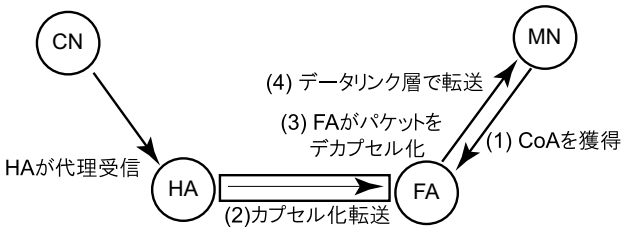

図 1: FA モード

MN の移動先に FA が存在しない場合は、MN 自身が FA となり、移動先で DHCP などの手段によって割り当てられた IP アドレスを CoA として HA に送信する (Co-located Care-of Address モード、図 2)[8]。

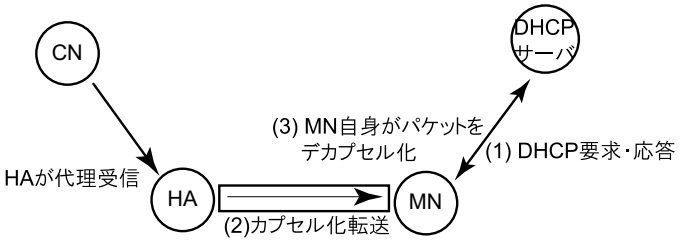

図 2: Co-located Care-of Address モード

従って、Mobile IP を利用するためには、移動端末に Home Address として利用する固定 IP アドレスを割り当てる必要があるため、そのような固定 IP アドレスが利用できない場合には Mobile IP は利用できない。

また、HA-FA 間のカプセル化通信による IP パケットの転送が Mobile IP を利用する上で必須となるため、ホームネットワーク内に HA を設置する必要がある。

さらに、NAPT が存在し、FA が NAPT 内部のネットワークにいる、もしくは FA が存在せず、MN の IP アドレスを CoA として使用する場合 (Co-located Care-of Address モード)、IP アドレスの変換機能のために MN が HA に送信する CoA は意味のないものとなってしまう。また、NAPT

[2]ここでは、HA の属するセグメントを意味する。

が転送するパケットは、ポート番号などの IP アドレス以外の識別子を持つパケット、すなわち TCP、UDP、ICMP パケット (Echo 及び Echo Request のみ[3]) に限られるため [4]、HA から送信される、CN の IP パケットをカプセル化した IP パケットは MN に転送することができない。従って、NAPT が FA の機能を持たなければ Mobile IP を利用することはできないということになる。従って、一つの IP アドレスを複数の機器で共有しながら利用する必要がある場合には NAPT に FA の機能が必須となる。

このように、Mobile IP を利用するには、ホームネットワーク内に HA が設置できることと、移動先に FA が存在する (FA モードを利用)、もしくは移動先に FA も NAPT も存在しない (Co-located Care-of Address モードを利用)、ということが条件となる。

2.2 トランスポート層における移動透過性

一方、トランスポート層における移動透過性の解決手法もいくつか提案されている。これは、TCP や UDP のソケットの持つ IP アドレスとポート番号を移動に応じて書き換えることによって、アプリケーションに対して透過的に、あたかも通信が再開しているかのように見せることが可能となる、というものである (図 3)。

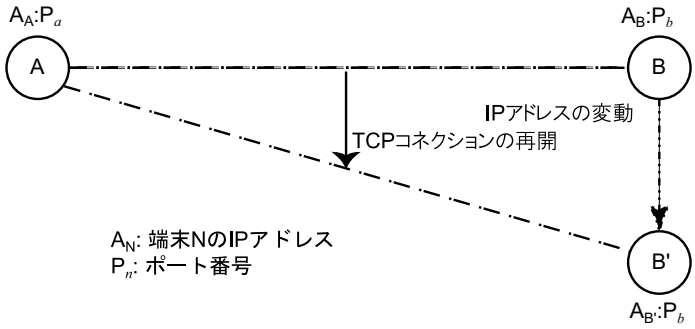

図 3: 端末移動時の TCP コネクション維持

Mobile TCP Socket[2] は、端末が移動したときに、移動先の IP アドレスを用いて TCP コネクションを確立し直し、アプリケーションに対しては通信が切断されていないように見せる仕組みを提案している。

TCP[5] は、2 つの端末がそれぞれポートを生成し、バッファを管理して再送制御を行ないながら、2 つのポート間でパケットを送受信するプロトコルである。すなわち、通信相手を識別するために送受信両側の IP アドレスとポート番号を用いるため、通信の途中でそれらのうち一つでも変化すると、通信が途切れてしまうこととなる。

TCP コネクションが切断されると、そのポートを使用している上位層のアプリケーションにはその旨が通知され、アプリケーションもまた通信を異常終了することとなる。従って、IP アドレスが変化しても、アプリケーションを拡張することなく通信を維持できるようにするためには、IP アドレスの変動を吸収してコネクションを維持するように TCP を拡張することとなる。

一般に、TCP/IP で通信を行なう際、アプリケーションはソケット API を経由して TCP/IP の各種機能を利用する。そこで、Mobile TCP Socket では、端末が移動して IP アドレスが変化して

[3]Echo 及び Echo Request メッセージは、IP アドレスに加えて、アプリケーション単位の識別子を利用するため、NAPT はこれをポート番号と同様に扱うことができる。

も、ソケット API では同じポートを利用することでアプリケーションの通信を再開できるようにするため、ソケット API と TCP インタフェースの間にモバイルソケット層 (MSL; Mobile Socket Layer) を挿入する (図 4)。

MSL は、ソケット API が使用するポートと一対一に静的に対応する仮想ポートを生成する。この仮想ポートは、TCP のポートと一対一に動的に対応し、端末の IP アドレスが変化したときに新たに TCP コネクションを確立しなおすことで通信を再開する。このとき、MSL は、ソケット API におけるポートと仮想ポートとの対応関係を維持することによって、下位層の IP アドレスやポート番号の変化を隠蔽するため、アプリケーションには同じポートで通信し続けているように見える。

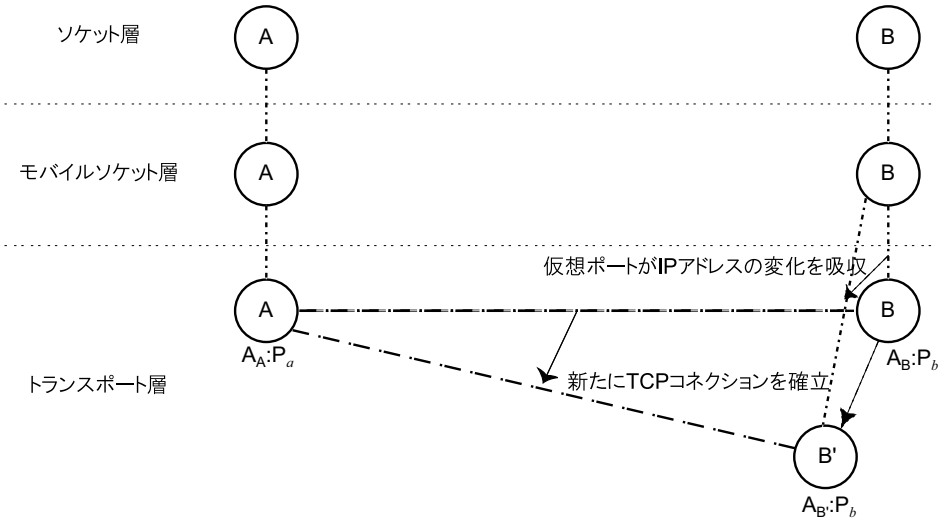

図 4: Mobile TCP Socket

しかし、この手法においては、移動前後で異なる TCP コネクションを利用するため、それぞれのコネクションが持っていたバッファが再利用されない。従って、TCP が再送制御の機能を持っているにもかかわらず、MSL においても TCP コネクションの切り替えに対処するための再送制御機能を別途持たなければならない。

これに対して、An End-to-End Approach to Host Mobility[3] では、端末の移動時に TCP コネクションが解放されないように、TCP を拡張している。

まず、通常の TCP の動作を表す状態遷移図を図 5 に示す [9]。`CLOSED` 状態から `ESTABLISHED` 状態へと続く矢印はコネクションの確立手順に対応する。コネクションが確立されているときは `ESTABLISHED` 状態となり、ほとんどのデータ転送は `ESTABLISHED` 状態で行なわれる。`ESTABLISHED` 状態から `CLOSED` 状態に戻るまでの矢印は、コネクションの解放手順に対応する。

TCP コネクションの確立は、確立要求を表わす SYN と、応答を表わす ACK を端末間で交換することによって行なわれる。SYN と ACK はいずれも TCP ヘッダ中の 1 ビットのフィールドに対応するフラグである。このとき、双方向で確立要求を交換して全二重通信を確立するため、双方向で SYN を交換するが、最初の SYN に対して応答するとき、SYN と ACK は一つのパケットにまとめて送信される (このパケットを一般に SYN, ACK パケットと呼ぶ)。これにより、TCP コネクションの確立手順は 3 段階で行なわれることとなる。この手順は一般に 3 ウェイハンドシェイクと呼ばれる。

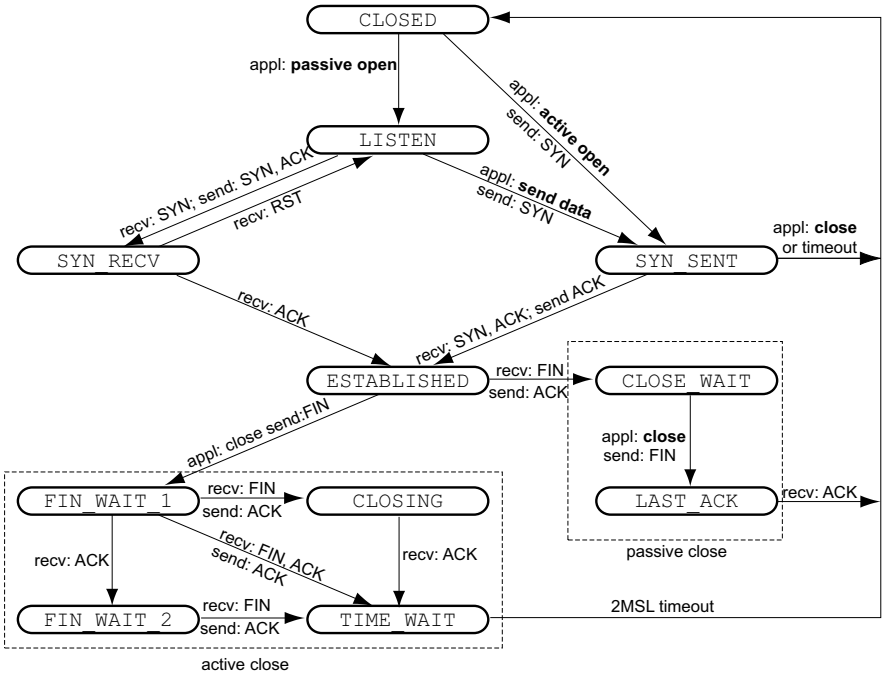

図 5: TCP の状態遷移図

TCP はデータの送受信が何バイト目まで完了したかを端末が互いに確認しあうためにシーケンス番号を交換する。シーケンス番号の単位はバイト (オクテット) である。データを送信する端末は、送信するデータの先頭バイトのシーケンス番号を TCP パケットのヘッダに記述してパケットを送信する。また、相手端末から次に予期する受信データのシーケンス番号を、ACK の確認番号として TCP パケットのヘッダに付与する。これによって、データの送受信が何バイト目まで正常に行なわれたかを相互に確認できるようになり、再送制御を可能にしている。

一方、シーケンス番号の初期値は、一般に 0 ではなく複雑な値に設定される。この初期シーケンス番号は、TCP コネクションの確立手順において端末間で交換し、以後データの送受信が発生する毎に値を増加させる。

初期シーケンス番号の交換も含めると、端末 A と端末 B の間での TCP コネクションの確立手順 (3 ウェイハンドシェイク) は次のようになる (図 6)。

1. A は B に SYN パケットを送信する。このとき、初期シーケンス番号を生成し、SYN パケットに付与する。

2. B は A から SYN パケットを受信すると、その初期シーケンス番号に 1 を足した値を ACK の確認番号として付与し、1. と同様にして初期シーケンス番号を生成して付与した SYN, ACK パケットを A に送信する。

3. A は B から SYN, ACK パケットを受信すると、その初期シーケンス番号に 1 を足した値を ACK の確認番号として付与した ACK パケットを B に送信する。

従って、端末の移動後に、移動先の IP アドレスやポート番号を通知するために、移動直後に通常の TCP コネクション確立手順と同様の 3 ウェイハンドシェイクを行なうことによって、TCP コ

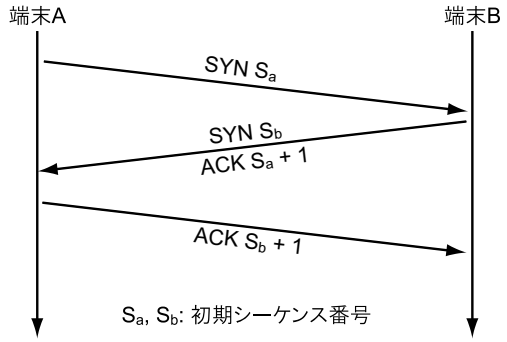

図 6: TCP コネクションの確立手順

ネクションを再開して端末間の通信を維持することを可能にする。このとき、互いのシーケンス番号が維持されるように初期シーケンス番号を与えれば、通信再開時に TCP が持つ再送制御機能によって、データの消失の無い通信の維持が可能となる。

しかし、ESTABLISHED 状態のソケットが、通信中の相手と違うソケットから SYN パケットを受信すると、通常は不正な SYN とみなして、確立要求を拒否するためリセットを表わす RST を送信する。従って、このままでは TCP コネクションを再開できないため、再開要求を表わす SYN と通常の SYN を区別するための手段が必要となる。

そこで、区別のために再開要求を表わす SYN には TCP オプション [5] として Migrate オプションが付与される。TCP オプションとは、TCP ヘッダの最後に 32 ビット単位で付与することが許可された任意のオプションである。Migrate オプションには、コネクションの再開回数及び端末の移動前の IP アドレス及びポート番号が記述され、再開要求が正しいかどうかを検査するために利用される。

Migrate オプションを付与した SYN パケットによって TCP コネクションの再開を可能とするために、TCP の動作が拡張されている。これによって、状態遷移図は図 7 のようになる[4]。

これらの手法は、経路上のノードは従来のままで、両端の端末にのみ TCP の拡張が必要となるため、Mobile IP に比べて導入の際に必要となる機器が非常に少ない。

2.3 端末の同時移動と NAPT を越えた移動

2.2 節で述べた手法においては、移動に関して次のような制約がある。

第一に、互いに通信する 2 つの端末のうち片方が移動している間にもう片方の端末が移動することによって、端末の移動する時間帯が重なる場合 (以後、これを端末の同時移動と呼ぶ)、TCP コネクションの維持は不可能である。これは、両端末が同時に移動すると、互いに相手の移動前の IP アドレス宛にしか自身の移動先を通知することができず、結果として互いに移動後の相手に移動先を通知することができないためである (図 8)。

第二に、NAPT を越えた移動時の通信再開は不可能な場合が多い。

NAPT(Network Address Port Translator) とは、複数の端末が限られた数の IP アドレスを共有して通信できるようにするために、変換したポート番号によってそれぞれの通信を区別し、多重化する装置である (図 9)[4]。

[4]TCP コネクションの解放処理は [3] において省略されており、本論文においてもそのまま引用した。

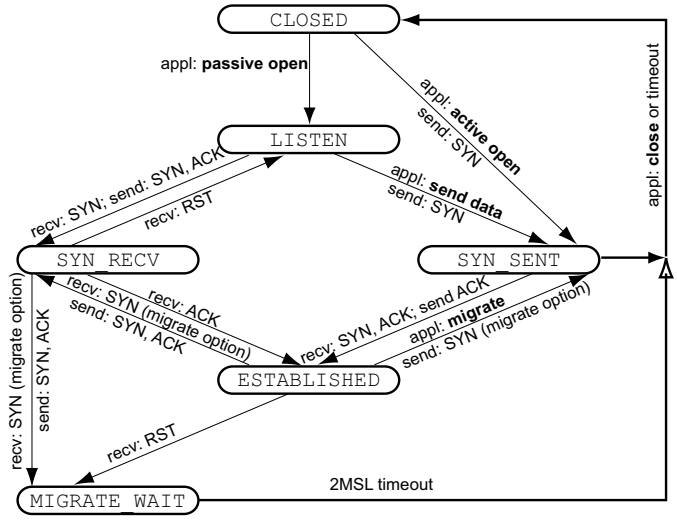

図 7: Migrate オプションによる TCP の状態遷移図の拡張

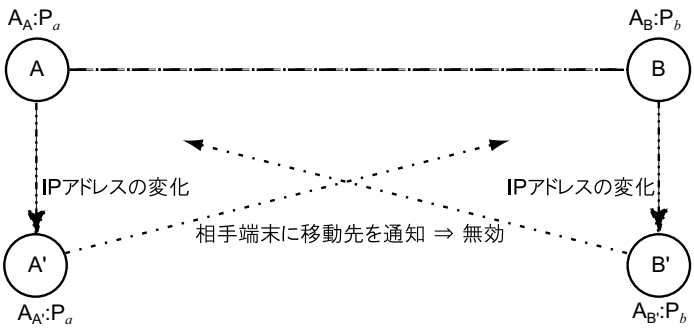

図 8: 端末の同時移動

以下、NAPT から見てグローバル IP アドレス空間に属するネットワークを NAPT 外 (外部) と呼び、プライベート IP アドレス空間の IP アドレスが割り当てられ、外側の端末にパケットを送信するときに、宛先 IP アドレスが NAPT に割り当てられたグローバル IP アドレスに変換されるネットワークを NAPT 内 (内部) と呼ぶこととする。

NAPT は NAPT 内の端末からの接続要求 (TCP の場合は SYN パケット) を受信し、外部に転送しようとするときに、外部と内部のアドレス変換の対応関係を設定し、この変換表に基づいて IP アドレスとポート変換を伴ったパケット転送を行なう。逆に、外部の端末から SYN パケットを受信した場合、通常は変換表に存在していない IP アドレスとポート番号から送信されたパケットであるため、NAPT はこれを無視するか、もしくは RST パケットの送信によってコネクションの確立を明示的に拒否する[5]。

従って、NAPT 外の端末が再開要求オプションを付与した SYN パケットを送信しても、その移動先 IP アドレス及びポート番号と、NAPT 内の端末との間の通信がそれまでに行なわれていないため、SYN パケットは NAPT の配下の端末には渡されない。従って、再開要求オプションを付与

[5]NAPT の動作を規定する RFC 3022[4] は、外部ネットワークの端末から SYN パケットを受信したときの応答を規定していない。外部からの SYN パケットに対する応答は NAPT の実装に依存する。

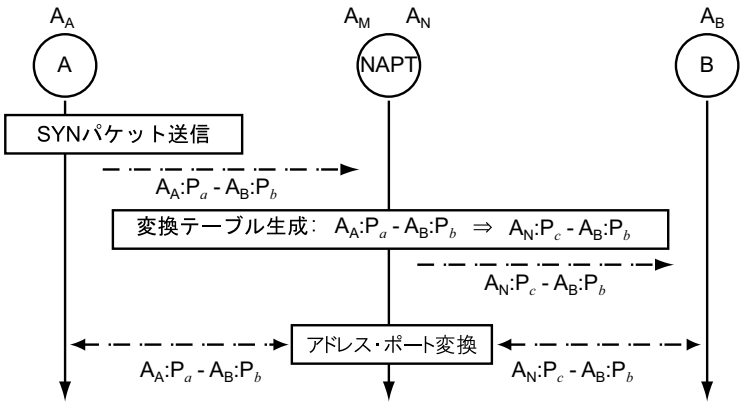

図 9: NAPT (Network Address Port Translator)

した SYN パケットは NAPT 内の端末から送信しなければならない。しかし、NAPT 内の端末は相手端末の移動先の通知を受けることも NAPT によって妨げられるため、NAPT 内の端末は SYN パケットの送信先を知ることができず、通信を再開することが不可能となる (図 10)。このように、NAPT によって TCP コネクションの確立方向が片方向に制限される性質を、以下では NAPT による確立の片方向性と呼ぶこととする。

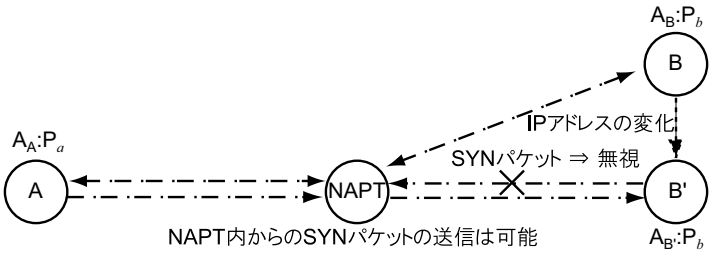

図 10: NAPT による確立の片方向性

また、互いに通信している両方の端末が別々の NAPT 内に移動した場合は、どちらの端末から送信された SYN パケットも相手端末に届かないため、仮に相手の移動先を知ることができたとしても、TCP コネクションの再確立が不可能となる (図 11)。

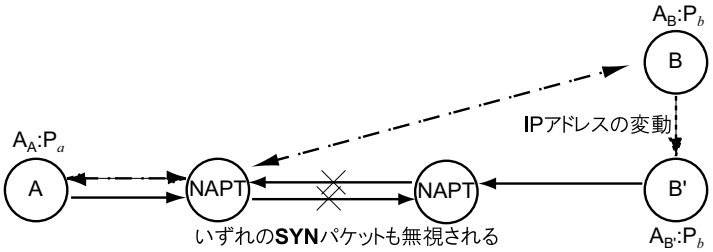

図 11: 別々の NAPT 内の端末における TCP コネクション再開の障害

これらの問題を解決するには、同時移動時の移動先の通知手法と NAPT 存在時における TCP コネクションの再開手順、両端末がそれぞれ異なる NAPT 内に移動したときにおける TCP コネクションの再開手法のそれぞれについて考察する必要がある。

第 3 章では、端末の同時移動や NAPT の存在によって相手端末の移動先を知ることができないときに、サーバを経由して端末の移動先を交換することによって TCP コネクションを再開するための手法、及び両端末が別々の NAPT 内に移動したときに通信を再開するための手法を提案する。

3 TCP コネクション維持プロトコル

本章では、第 2 章で述べた同時移動及び NAPT を越えた移動において、IP アドレスの変化を吸収して TCP コネクションの維持を行なうためのプロトコルについて説明する。

このプロトコルは、端末の移動に関する情報を交換する手順を与える移動先情報交換プロトコル、端末が移動したときにコネクションを解放せずに端末の移動先の IP アドレスとポート番号を用いてコネクションを再開するように拡張した TCP、両端末が別々の NAPT 内に移動したときに TCP コネクションを中継することよって擬似的に通信を維持する TCP コネクション中継プロキシの 3 つからなる。

3.1 移動先情報交換プロトコル

2.3 節で述べたように、端末の同時移動が発生した場合、もしくは NAPT 内の端末と通信している外部の端末が移動した場合には、どちらの端末も相手端末の移動先を知る手段がないために、相手端末に対して再開要求を付与した SYN パケットを送信することができず、TCP コネクションを維持することができないという問題がある。

従って、このような状況においても TCP コネクションを維持するためには、どちらか片方の、もしくは両方の端末の IP アドレスが変動したときに、相手端末の移動先を取得する仕組みが必要となる。

そこで、端末の移動に関する情報の交換を支援するサーバ (以後、移動情報管理サーバと呼ぶ) を経由して、端末の移動先の IP アドレスとポート番号を相手端末に通知するためのプロトコルとして、移動先情報交換プロトコルを定義する。

移動情報管理サーバは、端末がどのように移動したとしてもその移動先から通信可能となるようにするために、グローバル IP アドレスの空間に配置し、NAPT やファイヤウォールの影響を受けないようにする必要がある。また、サーバは同時に複数の端末ペアの移動に関する情報の交換を支援できることが望ましい。

3.1.1 移動先の特定に必要となる情報

移動前後において端末間の対応付けを維持するために必要となる情報は次のとおりである。

- 端末の移動直前における自端末 (src) と相手端末 (dest) の IP アドレス
- 移動前に src-dest 間で確立されている全 TCP コネクションのポート番号の対
- src の移動前後それぞれのポート番号 (変化したもののみ記述)
- src の移動先 IP アドレス

- ペア識別子 (後述)

移動情報管理サーバは、移動端末からこれらの情報を収集し、受信した情報を比較する。移動前に TCP コネクションを確立していた端末の組に該当する情報が見つかった場合、サーバはその端末のうちいずれかに対して相手端末の移動先の IP アドレスとポート番号を通知する。これによって、サーバから通知を受けた移動端末は、相手端末の移動後の IP アドレスに対して、再開要求を送信し、TCP コネクションを再開することが可能となる (図 12)。

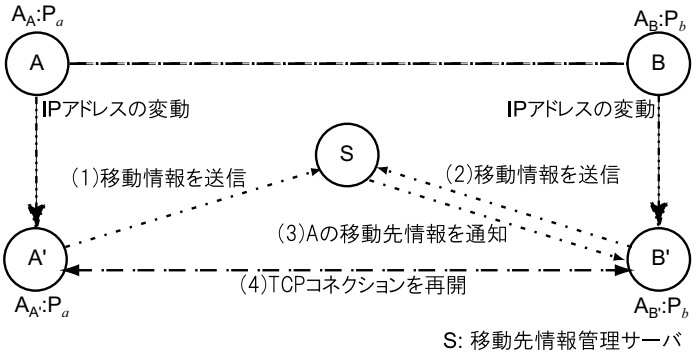

図 12: 移動情報の通知と TCP コネクションの再開

しかし、ある移動端末が移動前に使用していた IP アドレスを、移動後に別の端末が使用するような場合が発生すると、IP アドレスの重複によって、間違った相手端末と TCP コネクションの再開を指示してしまう恐れがある。この重複に対処するため、端末ペアの識別子を両端末の IP アドレスと乱数から生成する。この識別子をペア識別子と呼ぶ。それぞれの端末は通信開始時にペア識別子を生成して相手端末と共有し、サーバへ移動先を通知するときにペア識別子を付与することとする。ペア識別子は端末が移動するまでに端末間で交換し、同じ識別子を共有しなければならない。

以後、ペア識別子と移動前後の IP アドレスとポート番号をあわせて、移動情報と呼ぶこととする。

また、移動情報管理サーバが、TCP コネクションを再開すべき端末のペアを検出するためには、両方の端末が同じサーバに対して移動情報を送信しなければならない。従って、端末が移動する直前までに、相手端末との間で移動情報管理サーバの IP アドレスとポート番号、及びペア識別子を決定しなければならない。

ここで、確実に移動情報の交換を行なうために、ユーザが複数の移動情報管理サーバを指定した場合は、端末は全てのサーバに対して移動情報を送信する。それに対してサーバが端末にその相手端末の移動先を通知したとき、受信した端末は最初に受信した通知を採用するものとする。

なお、IP アドレスに加えてポート番号も移動情報として必要となるのは、3.3 節で述べる「TCP コネクション中継プロキシ」を利用する際にポート番号の変更が発生ためである。

以上により、移動情報の交換手順は、(1) 移動前の移動情報管理サーバとペア識別子の決定と、(2) 移動情報管理サーバを経由した移動情報の交換に分かれる。以下、(1) の手順を移動前情報交換、(2) の手順を移動後情報交換と呼ぶこととする。これらの情報交換は、TCP コネクション単位ではなく、端末-端末ペアを単位として行なう。

移動前情報交換は、端末が移動する直前までに完了すればよい。実際にいつ行なうかは実装、あるいはユーザによる指示のタイミングに依存する。例えば、2 端末間で最初に TCP コネクション

が一つ確立したときに自動的に実行する、あるいはユーザの指示があったときに実行する、といった方法が考えられる。

移動後情報交換を行なうタイミングは次の 2 通りがある。

- 端末の移動によって 3.2 節で述べる TCP コネクションの再開手続きが発生し、それが同時移動や NAPT による確立の片方向性によって失敗に終わったとき
- 端末が NAPT 内にいるために相手端末が移動してもその移動先の通知を受けることができないとき

後者については、実際に行なわれるタイミングは実装に依存する。例えば、一定時間おきに、あるいはユーザからの指示があったときにサーバに問い合わせを行なうという方法が考えられる。

3.1.2 NAPT の検出

NAPT 内の端末が外部の端末と通信する際、相手端末には NAPT によって変換された IP アドレスとポート番号が送信元のアドレス及びポートとして見えることとなる。従って、端末が移動した後で移動情報管理サーバに移動情報を送信したとき、サーバがどの端末間の通信を再開させるべきかを判定するには、移動前に端末が NAPT 内にいたかどうかを知っている必要がある。

また、NAPT 内の端末は、NAPT による確立の片方向性によって、相手端末の移動先の通知を受信することができない。従って、NAPT 内の端末が相手端末の移動先を知るためには、移動情報管理サーバへの問い合わせが必要となる。そのためには、移動端末自身が NAPT 内に移動したかどうかを検出する手段が必要となる。

通信を再開する端末ペアであるかどうかを移動情報によって判定するには、ペア識別子が同じものを選択するようにすればよい。以下では、ペア識別子が一致する 2 つの移動情報をサーバがそれまでに受信した移動情報の中から検索したときに、その送信元である 2 つの端末が移動前及び移動後に NAPT 内に移動しているかどうかを検出する方法を示す。

移動前の NAPT の存在を検出するには、移動後に両端末から移動情報管理サーバに通知される移動情報を互いに比較する。

端末 A, B がサーバに移動情報 A, B を送信し、ペア識別子によって通信を再開する端末ペアであることをサーバが確認したとき、サーバは 2 つの移動情報について次の要素を比較する。

1. 移動情報 A に記述された src の移動前アドレスと、移動情報 B に記述された dest の移動前アドレス:これらが異なる場合、A が NAPT 内にいたということになる。
2. 移動情報 B に記述された src の移動前アドレスと、移動情報 A に記述された dest の移動前アドレス:これらが異なる場合、B が NAPT 内にいたということになる。
3. 移動情報 A 及び移動情報 B に記述された dest の移動前アドレス:これらが同じ場合、端末 A, B は、移動前に 3.3 節で述べる TCP コネクション中継プロキシを利用して通信していたということになる。

以上 1, 2, 3. のいずれにも当てはまらなかった場合、サーバはペア識別子が偶然一致したものと判断して、受信した移動情報の相手端末に相当する移動情報を改めて検索しなおすこととする。

移動後の NAPT の存在を検出するには、次のようにする。

端末は移動後情報交換の際、移動情報管理サーバに移動情報を送信し、サーバとの間の TCP コネクションを解放せずに、サーバからの相手端末の移動先通知を待つ。このとき、移動情報を運ぶ

パケットのヘッダに記述された送信元 IP アドレス (以下、実際の送信元 IP アドレスと呼ぶ) は、もし端末が NAPT 内にいれば、NAPT によって端末自身が利用している IP アドレスと異なったものに変換されることになる。

従って、移動後情報交換において、移動情報管理サーバが端末から移動情報を受信したとき、移動情報に記述されている src の移動後の IP アドレスと実際の送信元 IP アドレスを比較して、異なっている場合はその端末が NAPT 内にいるということがわかる。

但し、NAPT によるポート番号の変換は、この方法では検出できない。そこで、Migrate-Permit オプション (3.2.2 節参照) によって相手端末が直接持っている (相手端末が NAPT 内にいる場合は NAPT によって変換される前の) ポート番号を獲得する。移動情報には、このポート番号を記述する。

3.1.3 移動前情報交換

移動前情報交換、すなわち移動端末間のサーバ指定とペア識別子交換は次の手順で行なわれる。

1. 端末 A と端末 B が通信しているときに、A がアプリケーションから指示を受けると、A は B との間に TCP コネクションを確立する。
2. 1. で確立した TCP コネクション上で、相手端末に次の情報を送信する。
   - 移動情報管理サーバの IP アドレスとポート番号
   - 端末の IP アドレス
3. B は、2. で送信された情報を受信すると、受信した相手端末の IP アドレスと自端末の IP アドレスと乱数から生成したペア識別子を A に返送する。A が NAPT 内にいることが検出された場合はその旨を併せて通知する。
4. A は 3. で送信された情報を受信すると、確認応答を返信する。B が NAPT 内にいることが検出された場合はその旨を併せて通知する。
5. この情報交換のために確立した TCP コネクションを閉じる。

端末間でサーバ指定とペア識別子交換が完了すると、端末の同時移動、ないし NAPT を越えた移動が発生したとき、2 つの端末がそれぞれサーバとの間で移動先情報を交換することによって、サーバが移動端末に対して相手端末の移動先を通知することができるようになる。

3.1.4 移動後情報交換における移動端末の動作

3.1.3 節で説明した移動前情報交換が完了した端末が移動したとき、端末は次のように動作する。

1. TCP においてコネクション再開手続き (3.2 節参照) が失敗したとき、ないしユーザやアプリケーションからの指示があったとき、端末は移動情報管理サーバとの間に TCP コネクションを確立して、移動情報を送信する。複数のサーバを利用するように設定されている場合は、全てのサーバに対して同じ情報を送信する。この TCP コネクションは、以下の手順においてサーバが端末に対して相手端末の移動先を通知できるようにするために、端末側から解放されるまで一定時間確立したままにする。

2. サーバから 1. で確立した TCP コネクションで相手端末の移動先を通知された端末は、相手端末の移動先に対して 3.2 節で述べる TCP コネクション再開手続を開始する。

3. 相手端末と TCP コネクションの再開が完了したとき、ないし一定時間内にサーバからの相手端末の移動先通知と、相手端末との TCP コネクションの再開のいずれも発生しないまま一定時間が経過したとき、移動情報管理サーバとの TCP コネクションを解放する。複数のサーバに移動情報を送信した場合は、一つでも移動先通知があったとき、全てのサーバとの TCP コネクションを解放する。

4. 端末移動によってサーバとの TCP コネクションが中断される場合は、3.2 節で述べる TCP コネクションの再開手続きが実行され、サーバとの TCP コネクションが維持される。このとき、移動した端末は維持されている TCP コネクション上で改めて移動情報をサーバに送信し、2. の手順に戻る。

3.1.5 移動後情報交換における移動情報管理サーバの動作

移動情報管理サーバは、上記の手順において送信される移動情報を、複数の端末から受信して蓄積する。移動情報の受信時に、既に蓄積されている移動情報の中に、その移動情報の送信元の相手端末に該当するものがあれば、通信の再開要求を相手端末に対して送信することが可能な端末を判別して、その端末に対して相手端末の移動先を通知する。

まず、ペア識別子が一致する移動情報を検索する。もし src の移動後 IP アドレスが、受信した移動情報に記述された src の移動前 IP アドレスと一致するものがあれば、同じ端末が連続して移動したものと判断して、受信した移動情報で上書きする。

次に、3.1.2 節で説明した方法によって、端末が移動前と移動後において NAPT 内にいるかどうかを調べる。

続いて、通信を再開すべきポート番号の対を検索する。このとき、NAPT によってポート番号が変換された場合は、変換前のポート番号が移動情報に記述されている。移動前に端末が NAPT 内にいたかどうかによって、TCP コネクションを再開するポート番号の対の選択規則が異なる。

- 端末 A, B が外部にいた場合、もしくはいずれかが NAPT 内にいた場合: 移動前において、A の移動情報に記述された (以下、移動情報 A の)src ポートと移動情報 B の dest ポートが一致し、かつ移動情報 A の dest ポートと移動情報 B の src ポートが一致する TCP コネクションを再開の対象とする。

- 端末 A, B の両方が NAPT 内にいた場合: まず、移動情報 A と移動情報 B の両方について、移動前の dest の IP アドレスが一致する場合、TCP コネクション中継プロキシ (3.3 節参照) を利用していたものと判断する。この場合は、移動情報 A と移動情報 B において両方の dest ポート番号が一致する TCP コネクションを再開の対象とする。

移動後の各端末の NAPT との関係に対して、サーバが相手端末の移動先を通知すべき端末の選択規則は次のようになる。

- 両者とも外部にいる場合: 後からサーバに移動情報を送信した方に、相手端末の移動先を通知する。

- 片方が NAPT 内にいる場合: NAPT 内にいる方に相手端末の移動先を通知する。

- 両方が NAPT 内にいる場合: 3.3 節で説明する TCP コネクション中継プロキシを利用することによって、TCP コネクションの再開が可能となる。従って、プロキシの IP アドレスとポート番号を移動先として両端末に通知する。

以上の規則によって端末に相手端末の移動先を通知するとき、サーバが端末に次の要素を含む情報を送信する。以下、これを移動先情報と呼ぶ。なお、ここでは移動先情報を受信する側の端末を A、その相手端末を B としている。

- ペア識別子
- B の移動先 IP アドレス
- 移動前に A-B 間で確立されている全 TCP コネクションのポート番号の対
- A の移動前後それぞれのポート番号 (変化したもののみ記述):
  移動情報 A に記述された src のポート番号 (移動後のポート番号が記述されている場合はそれで上書きする) を移動後のポート番号、移動情報 B に記述された dest のポート番号を移動前のポート番号として移動先情報に記述する。

このような移動先情報をサーバから受信した端末は、相手端末に対して TCP コネクションの再開を要求し、通信再開を試みる。

以上をまとめると、移動情報ならびに移動先情報の送受信手順は次のようになる (図 13)。

1. 端末 A, B は、どちらかが移動するまでに移動前情報交換を行なう。
2. 端末 A, B はそれぞれ移動したとき、もしくはアプリケーションからの指示があったときに、移動情報管理サーバに移動情報を送信する。このときサーバと端末との間に確立された TCP コネクションは、開放せずにそのまま保持する。
3. 移動情報管理サーバは、端末 A, B 両方から受信した移動情報に基づいて、移動前後における経路上の NAPT の有無を検出して、移動先情報を生成し、その送信先を選択する。
4. 3. の結果に基づいて、端末 A, B のいずれか (TCP コネクション中継プロキシ (3.3 節参照) を利用する場合は両方) に移動先情報を送信する。

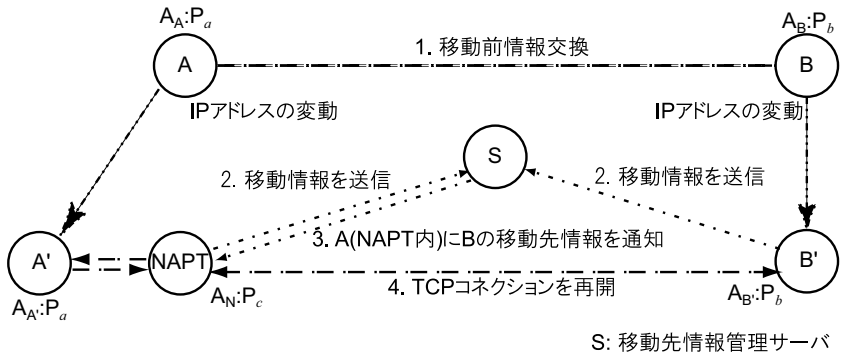

図 13: 移動先情報交換プロトコル

以上に示した手順によって、端末の同時移動や NAPT を越えた移動の後において TCP コネクションを再開し、通信を維持することが可能となる。TCP コネクションが再開された後は、次の手順によって移動情報管理サーバとの移動情報の交換を終了する。

1. 端末 A、B はそれぞれ移動情報管理サーバとの TCP コネクションを切断する。
2. 移動情報管理サーバは、端末 A、B から受信した移動情報を破棄する。

なお、端末がサーバに対して移動情報を送信したにもかかわらず、一定時間内にサーバからの相手端末の移動先通知もなく、TCP コネクションの再開も行なわれなかった場合も、上記の手順と同様にして移動情報の交換を終了する。

複数の移動情報管理サーバを利用する場合は、全てのサーバに対して TCP コネクションを確立して、移動情報の送信を行なう。

3.2 TCP の拡張による TCP コネクションの再開

2.2 節で述べたように、端末の移動後に TCP コネクションを再開することによってアプリケーション間の通信を維持するためには、IP アドレスとポート番号の変化を吸収する仕組みが必要となる。

本研究では、端末の同時移動及び NAPT を越えた移動の後にも TCP コネクションの再開を可能とするために、An End-to-End Approach to Host Mobility[3] の手法を拡張する。

なお、本手法はプロトコルのみの拡張によって、アプリケーションに変更を加えずに通信を透過的に維持することを目的とするため、TCP コネクションの維持についてのみ議論し、通信を持続させるための手段を持たない (コネクションレス通信である)UDP や ICMP などのプロトコルは扱わない。

3.2.1 TCP コネクションの確立と再開

2.2 節で述べた、手法 [3] による TCP の拡張をまとめると次のようになる。

- TCP コネクションの再開手順は、通常の 3 ウェイハンドシェイクと同じ手順によって行なう。
- 再開要求を表わす SYN パケットには、Migrate オプションを付与することによって、通常の SYN パケットと区別できるようにする。
- Migrate オプションを付与した SYN パケットの交換の際、移動前の IP アドレスを互いに確認した上で、ソケットが記憶する相手 IP アドレスとポート番号を更新して、相手の移動先と TCP コネクションを再開する。
- 初期シーケンス番号は、コネクションの再開後に移動前のシーケンス番号が維持されるように値を決定する。

端末の同時移動及び NAPT を越えた移動の後においても、TCP コネクションを再開できるようにするためには、さらに次のように TCP を拡張する必要がある。

- 再開要求を表わす SYN パケットが (相手端末も移動したことによって) 相手端末に届かなかったとき、アプリケーションから (移動先情報交換プロトコル (3.1 節) によって得られた) 相手端末の移動先 IP アドレスとポートを受け取ることによって再開要求を送信するようにする。

- 移動前の IP アドレスとポート番号を確認する際、NAPT による IP アドレスとポート番号の変換に対応できるようにする。

3.2.2 Migrate-Permit オプション

まず、TCP コネクション確立時に、互いに端末移動時に TCP コネクションの再開が可能であるかどうかを確認するための手段を TCP に追加する。

移動端末は、SYN パケットの送信時に、TCP オプションとして Migrate-Permit オプションを付与する。Migrate-Permit オプションは、原則として SYN フラグが立てられているパケット (SYN パケット、SYN, ACK パケットならびに MSYN パケット (3.2.3 節参照)) に付与されるが、移動先を通知する目的で SYN 以外のパケットに付与してもよい。

この SYN を受信した端末がもし Migrate-Permit オプションを解釈できる場合、その応答として送信する SYN, ACK パケットにも Migrate-Permit オプションを付与する。この手順によって、互いに IP アドレスが変化したときに TCP コネクションを再開することができるということを確認することが可能となる。

逆に、Migrate-Permit オプションを解釈できない端末に対して Migrate-Permit オプションを付与した SYN パケットが送信された場合、解釈できない TCP オプションは無視されるため、SYN パケットの応答として送信される SYN, ACK パケットには Migrate-Permit オプションが付与されない。従って、受信した SYN, ACK パケットに Migrate-Permit オプションが付与されていなかった場合は、端末移動時の TCP コネクション維持は諦め、従来の TCP と同じように動作するものとする。

ここで、TCP コネクションを再開する際、互いに TCP コネクションを確立していたポート同士であるかどうかを確認するため、TCP コネクションの再開要求を送信する際、両端末の移動前の IP アドレスとポートを Migrate-Permit オプションに付与する必要がある。

但し、NAPT が存在する場合、NAPT によって相手ポートに送信される IP アドレスとポート番号が変換されるため、単に IP パケットに記述された送信元の IP アドレスとポート番号を保存するだけでは、それを相手ポートに送信しても、相手が直接持っている IP アドレスとポート番号とは異なっているため、TCP コネクションが再開に失敗する原因となる。

そこで本手法では、Migrate-Permit オプションに自端末の IP アドレスとポート番号を付与することによって、端末が直接持つ (NAPT によって変換される前の) IP アドレスとポート番号、及び自端末に見えている (相手端末が NAPT 内にいる場合は NAPT によって変換された) 送信元 IP アドレスとポート番号を TCP コネクション確立時に交換できるようにする。TCP コネクションを再開する際、NAPT によって変換される前の IP アドレスとポート番号を再開要求に付与することによって、正しい相手ポートであるかどうかを確認することができる。

3.2.3 Migrate-Request オプション

端末の IP アドレスが変化したときに、移動前のコネクションの切断処理が行なわれずコネクションが存在したままとなっている場合、SYN パケットを相手ポートに送信すると、もしそのポートの状態が `LISTEN` ならば (すなわち、サーバアプリケーションがクライアントからの要求を受け付けるためのポートである場合は)、相手ポートは新しい TCP コネクションの確立を試みる。もしそれ以外の状態ならば SYN パケットは無視され、TCP コネクションは一定時間の待機後、タイムアウトとして解放されることとなる。また、送信先に NAPT が存在する場合は、NAPT がこれを

無視するため、タイムアウトによって TCP コネクションが解放されることとなる。

従って、SYN パケットをコネクションの再開要求として利用するには、通常の SYN パケットと区別できるようにする必要がある。そこで TCP コネクションの再開要求としての SYN パケットには、TCP オプションとして Migrate-Request オプションを付与することとする。以下、Migrate-Request オプションを付与した SYN パケットを MSYN パケットと呼ぶ。

Migrate-Request オプションは Migrate-Permit オプションと同様、SYN フラグが立てられているパケットのみに付与することができる TCP オプションである。

Migrate-Request オプションは、次の要素で構成される。

- TCP コネクション再開回数 ($ReqNo$)
- 移動前の相手端末の IP アドレスとポート番号
- コネクション識別子 $I(A_{src}, P_{src}, A_{dest}, P_{dest})$
  - A_{src}, P_{src}: 相手端末に見えていた移動前の自端末の IP アドレスとポート番号
  - A_{dest}, P_{dest}: 相手端末が直接持っていた移動前の相手端末の IP アドレスとポート番号
    コネクション識別子の引数は、3.2.2 節で述べた Migrate-Permit オプションで事前に相手端末から受け取った IP アドレスとポート番号を利用する

コネクション識別子 $I(A_{src}, P_{src}, A_{dest}, P_{dest})$ は、NAPT に変換される前後の両端末の IP アドレス及びポート番号の確認のために利用する。ここでは、TCP オプションの長さ制限[6]を考慮して、直接 $A_{src}, P_{src}, A_{dest}, P_{dest}$ を相手端末に送信する代わりにこれを用いて Migrate-Request オプションの長さを節約する。

3.2.4 migrate コマンド

MSYN パケットの送信のために、TCP における上位層とのインタフェースとして新たに **migrate** コマンドを追加する。**migrate** コマンドはあるコネクションに関して MSYN パケットを相手ポートに送信して TCP コネクションの再開を行なうためのコマンドで、例えば自端末の IP アドレスが変化したときや、相手端末の移動先を通知されたときに利用する。

migrate コマンドによって自端末の IP アドレスを更新するとき、移動前の IP アドレス A'_{src} とポート番号 P'_{src}、及び自端末に直接見えている相手端末の IP アドレス A'_{dest} とポート番号 P'_{dest} を保存する。これは、コネクション識別子 I によって MSYN パケットが正当なものであるかどうかを評価するために利用する。

コネクション確立時に Migrate-Permit オプションを交換できなかったポートについては、コネクションの再開は行なわない。

上位層から **migrate** コマンドが発行されたとき、まず該当するポートは移動先の相手ポートに MSYN パケットを送信する。このとき、間違った相手ポートに届いた場合は、次のような結果となる。

- 間違った相手が Migrate-Request オプションを解釈可能な場合: TCP コネクションの確立拒否のため、RST パケット (コネクションを強制的に解放するときに利用される) が送信される。

[6]TCP ヘッダの長さは 60 バイトに制限されている。このうち TCP オプション以外のヘッダ要素が 20 バイト使用しているため、TCP オプションに利用可能な部分は 40 バイトに制限される。

- 間違った相手が Migrate-Request オプションを解釈不可能な場合: MSYN パケットを受信したポートが LISTEN 状態の場合、新しいコネクションの確立のため、(Migrate-Request オプションのない)SYN, ACK パケットが送信される。LISTEN 状態でない場合は、無視される。

ここで、TCP 自身による再送制御が、コネクション再開後に正しく機能するようにするためには、シーケンス番号は初期化せずに、移動の前後でつじつまが合うように適切な値を設定しなければならない。すなわち、TCP コネクション再開後、最初に送信する TCP パケットのシーケンス番号 (SND.NXT) を、相手端末から ACK パケットによる応答が返されていない先頭のパケットのシーケンス番号 (SND.UNA) と一致させなければならない。

SYN パケットの次に送信する ACK パケットのシーケンス番号、すなわち SYN, ACK パケットの ACK の確認番号は、SYN パケットの初期シーケンス番号に 1 を足した値となる [5]。従って、シーケンス番号のつじつまを合わせるためには、MSYN パケットの初期シーケンス番号は SND.NXT - 1 = SND.UNA - 1 とする。

以上の議論は、MSYN に対する応答の SYN, ACK パケットにおける初期シーケンス番号にも適用される。

MSYN パケットが送信され、相手ポートに MSYN パケットが届くと、相手ポートはまず TCP コネクションの再開の対象となるポートを次のようにして検索する。

- 自端末のポートは、パケットの宛先ポートとする (もし自端末が移動した場合は、**migrate** コマンドによって既に自端末の IP アドレスの書き換えが完了しているものと考える)。
- 相手端末のポートは、Migrate-Request オプションで指定された IP アドレス及びポート番号とする。

相手ポートは送信されてきた MSYN パケットが正しい Migrate-Request オプションであるかどうかを次のようにして評価する。

- $ReqNo$ が自身の記憶しているものと等しくなければ不正な MSYN パケットとみなす。
- 事前に Migrate-Permit オプションによって受け取った $A'_{src}, P'_{src}, A'_{dest}, P'_{dest}$ から $I(A'_{dest}, P'_{dest}, A'_{src}, P'_{src})$ を計算し、受信した $I(A_{src}, P_{src}, A_{dest}, P_{dest})$ と一致しない場合は不正な MSYN パケットとみなす。

これによって正しい MSYN パケットであると評価した場合、MSYN パケットの生成手順と同様にして Migrate-Request オプションを付与した SYN, ACK パケット (以下、MSYN, ACK パケットと呼ぶ) を生成する。なお、ACK の確認番号は通常どおり受信した MSYN パケットの送信シーケンス番号に 1 を足した値とする。送信シーケンス番号の生成については MSYN パケットの場合と同様である。

この MSYN, ACK パケットが返送されると、これを受信した端末は上記の MSYN, ACK パケットの場合と同様の手順で処理し、正しい MSYN, ACK パケットであれば ACK パケットを送信する。

以上の手順により、TCP コネクションは再開され、移動によって中断されていたパケットの送信が行なわれる。

なお、次に端末の移動が発生したときに TCP コネクションの再開が同じように実行できるようにするため、ここで交換される MSYN パケット及び MSYN, ACK パケットにも Migrate-Permit オプションを付与することによって、NAPT に変換される前の IP アドレスを改めて相手端末に通知する。

3.2.5 コネクション解放の抑制

端末の IP アドレスが変化した際、MSYN パケットを送信したときに、同時に相手端末の IP アドレスも変化していると、MSYN パケットが相手端末に届かずにタイムアウトによって TCP コネクションの再開に失敗してしまう。

また、TCP コネクション上で通信を行なっているときに相手端末の IP アドレスが変化し、移動前の IP アドレスを他の端末が使用した場合、間違った相手にパケットが届いてしまい、その応答として RST パケットが送信され、TCP コネクションが切断されてしまう。

このとき、TCP コネクションを解放せず、上位層からもう一度 **migrate** コマンドが発行されるのを待つようにすると、TCP コネクションの再開をやり直すことができる。そのため、MSYN パケットに対する MSYN, ACK 待ちにおいてタイムアウトになった場合、及び TCP コネクションが確立しているときに RST パケットを受信した場合、直ちに TCP コネクションを解放せずに、上位層から **migrate** コマンドが呼び出されるのを待つこととする。一定時間待機して、MSYN パケットを受信できず、かつ **migrate** コマンドも発行されなかった場合は、TCP コネクションの再開を諦めてコネクションを閉じる。

3.2.6 TCP の状態遷移機械の拡張

以上の議論を踏まえ、TCP コネクション再開が可能となるためには、TCP を次のように拡張する必要がある。

- 相手端末の移動が原因で発生する TCP コネクションの解放を防止する。
- 上位層からの要求によって、MSYN パケットを送信して TCP コネクションの再開処理を実行する。

これらを可能とするために、TCP の状態遷移機械を図 14 のように拡張する。なお、簡単のためコネクションの解放処理に該当する状態遷移は省略した。

`MIG_SENT` 状態は、MSYN パケットを送信し、応答の MSYN, ACK パケットを待機している状態である。相手端末の移動などの原因によって、MSYN, ACK パケットが一定時間内に受信できなかったときは、次に示す `MIG_WAIT` 状態に移る。

`MIG_WAIT` 状態は、相手端末からの MSYN パケット及び上位層からの **migrate** コマンドの呼び出しを待機している状態である。この状態において、MSYN パケットを受信したときは MSYN, ACK パケットを返送して `SYN_RECV` 状態に移り、上位層から **migrate** コマンドが発行されたときは、MSYN パケットを送信して上記の `MIG_SENT` 状態に移る。一定時間内に MSYN パケットを受信できなかった場合は、`CLOSED` 状態に移り、コネクションを閉じる。

この状態遷移図に従うと、端末が移動したときに TCP は次のように動作する。

1. `ESTABLISHED`、`FIN_WAIT_1`、`MIG_WAIT` のうちいずれかの状態のときに上位層から **migrate** コマンドが発行されると、該当する TCP コネクションが事前に Migrate-Permit オプションのやり取りによって再開可能であるとわかっている場合、相手端末の移動先に MSYN パケットを送信し、`MIG_SENT` 状態に移る。
2. 自端末から相手端末までの経路上に NAPT が存在する場合、MSYN パケットの送信によって、相手端末の移動先と自端末との間の対応関係が NAPT の変換テーブルに新たに追加され、同時に NAPT を経由して相手端末から自端末へのパケットを送信することが可能となる。

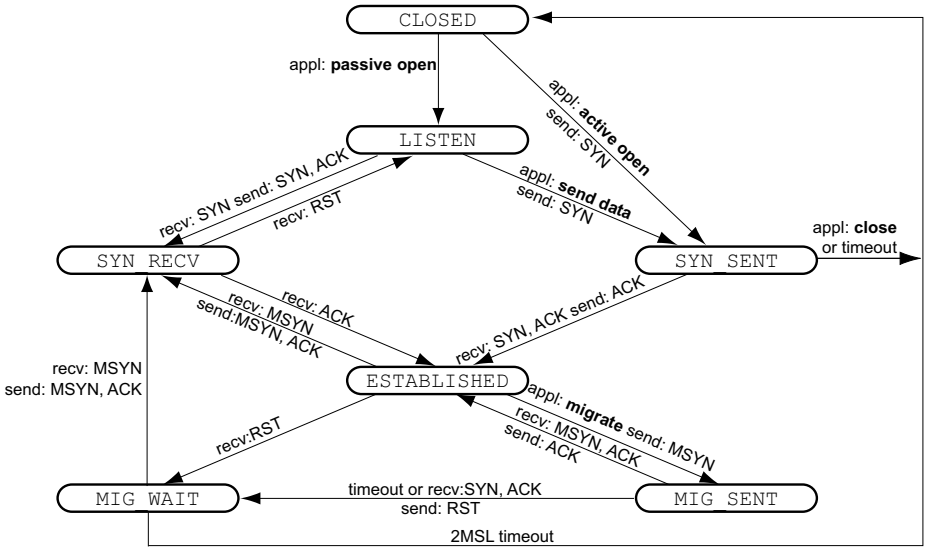

図 14: 拡張 TCP の状態遷移図

3. 相手端末が移動していない、かつ NAPT 内ではない場合、相手端末は 1. で送信された MSYN パケットを受信する。

4. 3. で MSYN パケットを受信した相手端末は、コネクション識別子とシーケンス番号、$ReqNo$ の検証を行ない、全て正しければ、1. と同様の手順で MSYN, ACK パケットを生成して送信し、`SYN_RECV` 状態に移る。

5. 4. で送信された SYN, ACK パケットを受信した端末は、4. と同様にしてコネクション識別子 I とシーケンス番号、$ReqNo$ を照合・検証し、全て正しければ ACK パケットを返送し、`ESTABLISHED`、あるいは `FIN_WAIT_1` 状態に戻る。このとき、$ReqNo$ に 1 を加算し、次回の TCP コネクション再開処理が発生したときにこの $ReqNo$ の値を使う。

6. 5. で送信された ACK パケットを受信した相手端末は、`ESTABLISHED`、あるいは `FIN_WAIT_1` 状態に戻る。これで TCP コネクションの再開が完了し、コネクションが維持されたこととなる。このとき、$ReqNo$ に 1 を加算し、次回の TCP コネクション再開処理が発生したときにこの $ReqNo$ の値を使う。

7. 3. に対して、相手端末が移動した、もしくは NAPT 内にいる場合、MSYN パケットは相手端末に届かない。この場合、一定時間の待機後、`MIG_WAIT` 状態に移り、1. に戻る。

8. `MIG_WAIT` 状態において一定時間内に MSYN を受信せず、**migrate** コマンドも呼び出されなかった場合、TCP コネクションを閉じる。

また、`ESTABLISHED` 状態のときに RST パケットを受信すると、`MIG_WAIT` 状態に移り、相手端末からの MSYN パケットの送信及び上位層からの **migrate** コマンドの発行を待機する。これは、相手端末の IP アドレスの変化が原因で間違った相手端末に届いたときに、エラーとして扱われて RST パケットを返送された場合、直ちに TCP コネクションを閉じるのではなく、移動後の相手端末とのコネクションの再開を試みることを目的としている。

TCP の状態遷移機械における [3] との大きな相違点は、MSYN パケットの送信に対して何の応答も返ってこなかった場合、すぐにはコネクションを解放せず、もう一度コネクションの再開を試みるという点である。

例として、端末 A、B の同時移動が発生した場合について、この拡張 TCP における TCP コネクションの再開手順を図 15 に示す。ここで、appl: **migrate** が発生するタイミングは、例えば 3.1 節で述べた移動先情報交換プロトコルによって、端末 A が移動情報管理サーバから端末 B の移動先を通知されたときに相当する。

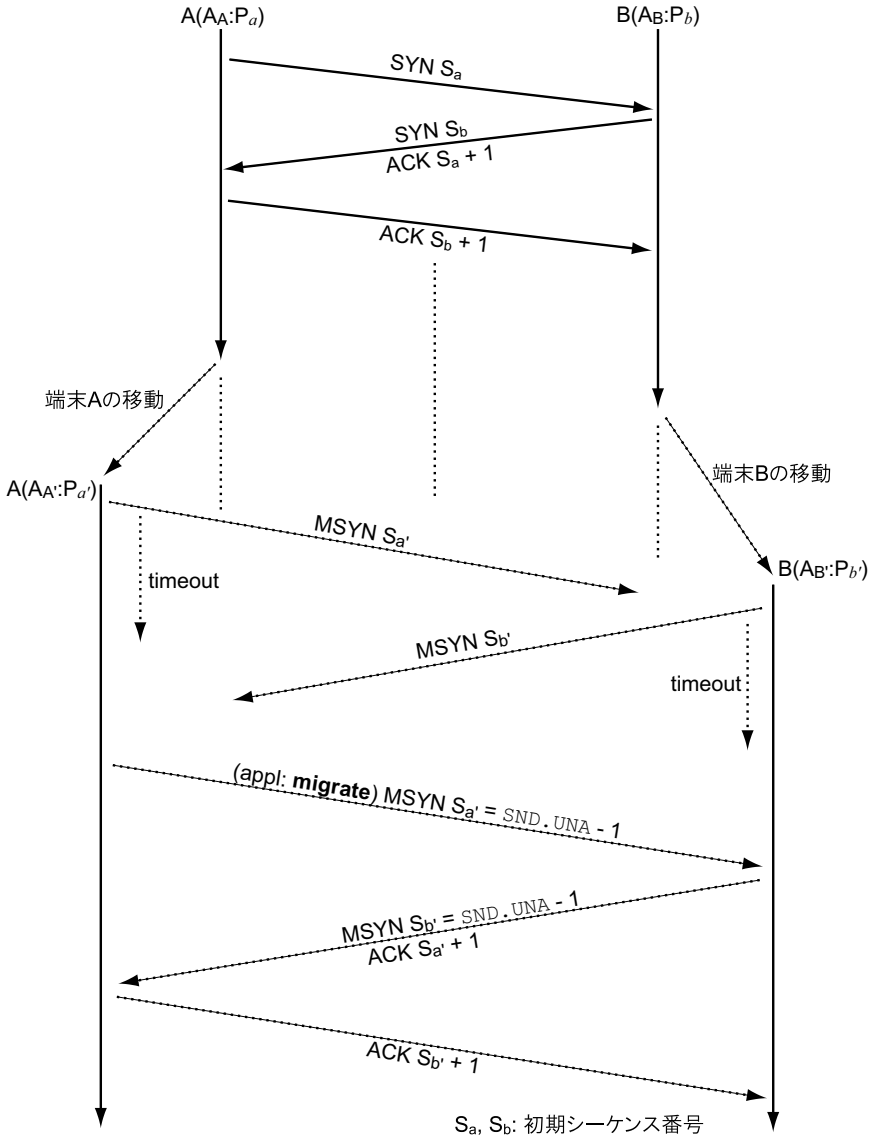

図 15: TCP コネクションの再開

3.3 TCP コネクション中継プロキシ

2 つの端末が互いに別々の NAPT 内に移動した場合、いずれの端末から送信された SYN パケットも、経路上において相手端末の手前に存在する NAPT によって遮られて相手端末まで届かなくなるため、このままでは TCP コネクションの再開は不可能となる。このような移動が発生した場合においても TCP コネクションを維持するには、NAPT 内にいる 2 つの端末の通信を外部で中継する仕組みが必要となる。そこで、両端末から MSYN パケットを受信して TCP コネクションを中継するプロキシを用意する。このように、中継の対象となる両方の端末から確立要求を受信することによって機能するプロキシを対称型プロキシと呼ぶこととする。

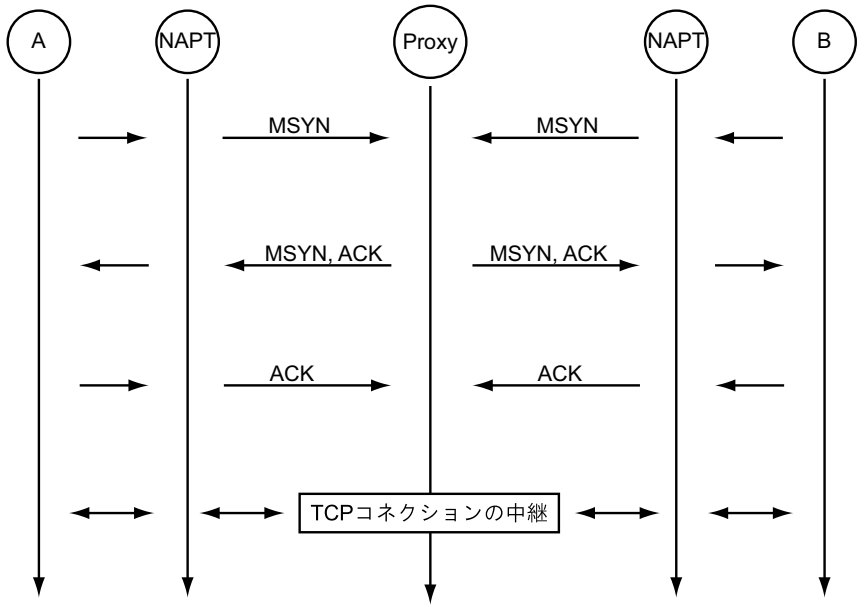

図 16: 対称型プロキシ

3.3.1 対称型プロキシによる TCP コネクションの中継開始

通信中の 2 つの端末が移動によって両方とも NAPT 内に移動した場合、TCP コネクションを再開するためには、対称型プロキシによって中継する必要がある。

3.1 節で示した移動先情報交換プロトコルによって移動端末に移動先通知を行なう際、3.1.2 節で示した方法によって両端末が別々の NAPT 内にいることがわかったとき、移動情報管理サーバは両方の端末に対して、互いの IP アドレスの代わりに非移動端末である対称型プロキシサーバを移動先として通知することで、通信の維持を実現する。

移動情報管理サーバは移動端末に移動先通知を行なう前に、どの TCP コネクションを中継させるかを対称型プロキシサーバに指示し、コネクション毎に中継に利用するポート番号の割り当てを受ける。ここで得られたポート番号とプロキシサーバの IP アドレスを端末の移動先として移動先情報を生成し、両端末に送信することによって、TCP コネクションがプロキシサーバを経由して再開することが可能となる。このとき、SIP[6][7] のプロキシモードと同様に、端末にプロキシに対応させるための特別な処理を組み込む必要はない。

以下、対称型プロキシサーバの中継による TCP コネクションの再開手順を示す。

1. 両端末が別々の NAPT 内にいることを検出した移動先情報管理サーバは、中継する TCP コネクション毎にプロキシのポートを確保するために、中継の対象となる両端末の IP アドレスとポート番号をプロキシに通知する。このとき通知する IP アドレスは、NAPT によって変換される前の IP アドレスを通知する。
2. プロキシは移動先情報管理サーバから TCP コネクション中継用のポートの確保を指示されたとき、ポートを確保し、中継の対象となる IP アドレスならびにポート番号と関連付ける。確保したポート番号はサーバに通知する。
3. 移動情報管理サーバは、プロキシの IP アドレスと 2. で確保されたポート番号の組を移動先としてそれぞれの移動端末に通知する。
4. 移動先通知を受信した両端末は、プロキシサーバに対して MSYN パケットを送信する。
5. プロキシサーバは、4. で端末から送信された 2 つの MSYN パケットの到着を待つ。MSYN パケットがコネクション中継の対象として正しいものであるかどうかは、(NAPT によって変換された) 送信元の IP アドレスとポート番号ではなく、Migrate-Permit オプションに記述された送信元 IP アドレスとポート番号によって判定する。
6. プロキシサーバは、両方の端末から MSYN パケットを受信した後、次の手順で 2 つの SYN, ACK パケットを生成し、返送する。
   - 片方の MSYN パケットのシーケンス番号に 1 を足したものを、その端末に返送する MSYN, ACK パケットの ACK の確認番号とする。
   - 片方の MSYN パケットのシーケンス番号を、もう片方の端末に返送する MSYN, ACK パケットのシーケンス番号とする。
   - 片方の MSYN パケットに含まれる Migrate-Permit オプション及び Migrate-Request オプションは、そのままもう片方の MSYN, ACK パケットに付与する。
     このようにすることによって、端末移動後にプロキシを中継しない TCP コネクションの再開を可能とする。
7. 3. で送信された MSYN, ACK パケットを受信した両端末は、ACK パケットを返送する。
8. プロキシサーバは、4. で送信された ACK パケットを両方とも受信したら、以後プロキシによる TCP コネクションの中継が実行される。

なお、プロキシサーバは 3. の手順において、Migrate-Permit オプションに含まれる送信元 IP アドレスとポート番号、Migrate-Request オプションに含まれるコネクション識別子を記憶する。これは、3.3.2 節で説明する、TCP コネクション中継時に端末が移動したときのコネクションの再開処理のために利用する。

3.3.2 プロキシによる TCP コネクションの中継

前節に示した手順によって、NAPT 内にいる 2 つの端末がプロキシサーバの中継によって TCP コネクションを再開することができる。

この後、対称型プロキシサーバは次に示す方法で端末 A, B 間で送受信される TCP パケットの受け渡しを行なう。

- A からプロキシサーバに送信されたパケットは、宛先を B、送信元をサーバ自身に書き換え、B に送信する。このとき、サーバは ACK の確認番号を記憶する。簡単のため、この ACK の確認番号を A の ACK 確認番号と呼ぶ。
- B からプロキシサーバに送信されたパケットは、宛先を A、送信元をサーバ自身に書き換え、A に送信する。簡単のため、この ACK の確認番号を B の ACK 確認番号と呼ぶ。

3.3.3 コネクション中継中の端末の移動

プロキシが TCP コネクションを中継している間に端末の IP アドレスないしポート番号が変化すると、3.2 節で示した手順によって TCP コネクションの再開処理が開始される。このとき、端末は直接の通信相手である対称型プロキシサーバに MSYN パケットを送信する。

プロキシサーバは端末の移動先から送信された MSYN パケットを受信すると、まず Migrate-Permit オプションに記述された送信元 IP アドレス及びポート番号と MSYN パケットの送信元 IP アドレス及びポート番号を比較し、NAPT の存在を検査する。

もし MSYN パケットを送信した端末の移動先が外部である場合は、MSYN パケットを無視し、その相手端末に RST パケットを送信し、プロキシによるコネクションの中継を中断する。これによって、両方の端末が移動情報をサーバに送信することとなり、この後、3.1 節で示した手順に従って TCP コネクションを再開することが可能となる。

MSYN パケットを送信した端末の移動先が NAPT 内である場合、引き続きプロキシサーバによる TCP コネクションの中継が必要となる。ここで、端末 A, B 間の TCP コネクションをプロキシサーバ P が中継しているときに B が移動した場合、プロキシサーバと移動端末の間のコネクション再開手順は次のようになる。

1. B は P に MSYN パケットを送信する。
2. P は MSYN パケットの内容を次の規則によって正しい MSYN パケットかどうかを評価する。
   - 移動前の IP アドレスとポート番号、コネクション識別子は、3.2.3 節と同様にして正しいかどうかを評価する。
   - シーケンス番号は、MSYN の (初期) シーケンス番号が A の ACK 確認番号から 1 を引いた値と等しければ、正しいものとして評価する。
3. 2. において正しい MSYN パケットであると評価したとき、プロキシサーバは 3.2.3 節と同様に MSYN, ACK パケットを生成して B に送信する。このとき、初期シーケンス番号には A の ACK 確認番号と同じ値を代入する。
4. P は B から ACK パケットを受信すると、TCP コネクションの中継を再開する。

このとき、A に対して B の移動先を通知するのは、A と B がプロキシによるコネクションの中継を介さずに TCP コネクションを再開しようとすると、A が記憶する B の移動前 IP アドレスとポート番号が古いままとなっているために、結果として TCP コネクションの再開は失敗することを防ぐためである。

従って、P が B から ACK パケットを受信した後、TCP コネクションの中継を再開する前に A に B の移動先を通知しなければならない。

P は A に B の移動先を通知するために、B の移動先から生成した Migrate-Permit オプションを付与した、データ部の長さが 0 バイトの TCP パケットを A に送信する。このパケットのシーケンス番号は A の ACK 確認番号と同じ値とし、ACK の確認番号は B のシーケンス番号と同じ値とする。A はこのパケットを受信することによって B の移動先を知ることができる。

3.3.4 プロキシ中継の終了

プロキシ中継の終了は、3.3.3 節で説明したように端末の外部への移動によって終了する場合と、通常の TCP コネクションの終了と同様に FIN パケットとそれに対する ACK パケットの交換によって終了する場合の 2 通りある。

前者については、コネクション中継のために割り当てられたポートを単純に解放して、コネクションの中継を中断する。

後者については、両端末から FIN パケットに対する ACK パケットまで、3.3.1 節の手順 5. で示した方法によってパケットの中継を続ける。この 2 つの ACK パケットの中継が完了したときに、中継用に割り当てられたポートを解放し、中継を終了する。

3.4 TCP コネクション維持のための制御手法

3.4.1 モジュール設計

3.1 節と 3.2 節、3.3 節で述べたプロトコルを組み合わせて利用することによって、NAPT を越えたあらゆる移動が生じた場合においても、端末の移動後に TCP コネクションを再開することが可能となる。

まず、移動端末のモジュール構成は次のようになる。

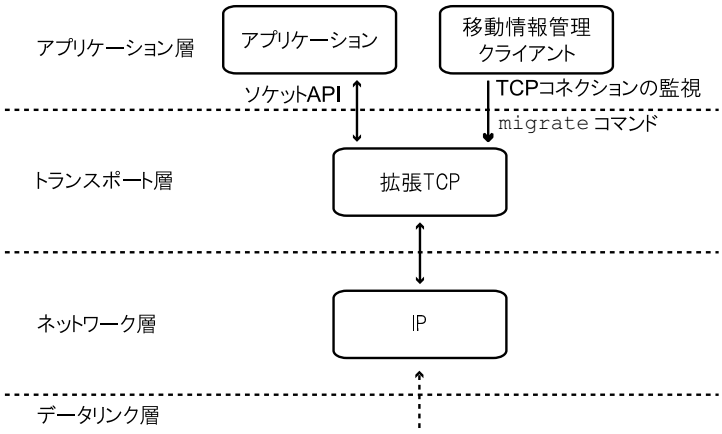

図 17: 移動端末のモジュール構成

移動情報管理クライアントは、次の役割を果たす。

- 端末の IP アドレスと TCP コネクションを監視する。

- 次の条件が発生したとき、該当する TCP コネクションに対して **migrate** コマンドを発行する。
  - 端末の IP アドレスが変化したとき
  - 移動情報管理サーバから相手端末の移動先情報を受信したとき
- 次の条件が発生したとき、移動情報管理サーバに移動情報を送信する。
  - TCP コネクションが `MIG_WAIT` 状態に移ったとき
  - ユーザからの指示があったとき

拡張 TCP は、3.2 節で説明した TCP コネクションを再開できるように拡張した TCP である。

アプリケーション及び IP 以下の層は変更する必要がない。また、アプリケーションに提供されるソケット API も、従来のものをそのまま利用する。

3.1 節で述べたように、端末の移動後に TCP コネクションを再開する手順は、それぞれの端末が NAPT 内にいるかどうかによって異なる。従って、移動先情報管理サーバが NAPT の存在を検出することによって、状況を判断してそれに応じたコネクション再開手順を制御する必要がある。

NAPT の存在は、3.1.2 節で述べた方法によって検出できる。移動先情報管理サーバは、2 つの移動端末のうちどちらが NAPT 内にいるかということを基準として、どちらの移動端末に対して相手端末の移動先通知を行ない、MSYN パケットの送信を指示するべきかを判断することができる。

3.4.2 両端末が外部にいる場合の TCP コネクション再開手順

両端末が外部にいる場合の TCP コネクションの再開手順は次のようになる。

片方の端末のみの移動の場合は、3.2 節で示した手順によって、拡張 TCP の機能のみによって TCP コネクションを再開することができる。

両方の端末が同時に移動した場合は、3.1 節で示した移動先情報交換プロトコルを利用して、移動情報の交換を行なう必要がある。この場合における TCP コネクション再開手順は次のようになる。

1. 両端末はそれぞれ移動情報を移動情報管理サーバに送信する。
2. 移動情報管理サーバは、両端末から移動情報を受信すると、後に受信したほうの端末に対して相手端末の移動先情報を送信する。
3. 2. で送信された移動先情報を受信した端末は、相手端末の移動先に MSYN パケットを送信し、TCP コネクションの再開を実行する。

3.4.3 片方の端末が NAPT 内にいる場合の TCP コネクション再開手順

片方の端末が NAPT 内にいる場合の TCP コネクション再開手順は次のようになる。

片方が NAPT 内に移動し、もう片方は移動せずに外部にいる場合は、NAPT 内の端末が MSYN パケットを送信することとなり、この MSYN パケットは外部の相手端末に届く。従って、3.2 節で示した手順、すなわち拡張 TCP の機能のみによって TCP コネクションを再開することができる。

片方が外部に移動し、もう片方が移動せずに NAPT 内にいる、もしくは NAPT 内に移動した場合は、いずれの端末から送信された MSYN パケットも相手端末に届かない。従って、3.1 節で示し

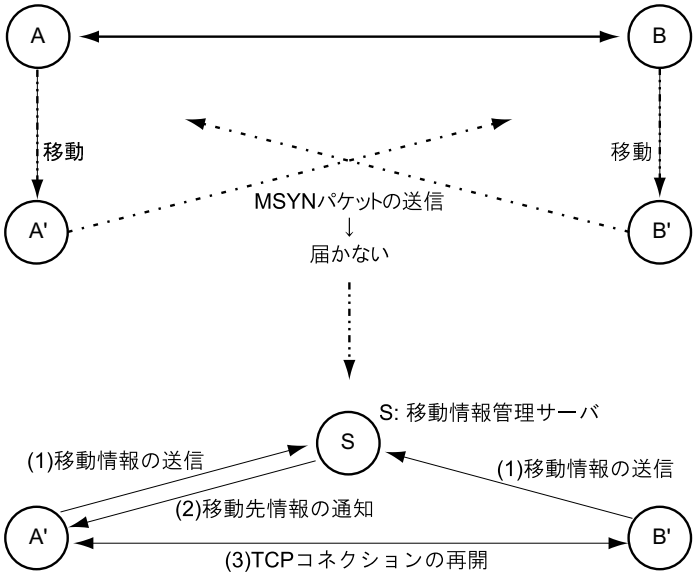

図 18: 外部における同時移動と TCP コネクションの再開

た移動先情報交換プロトコルを利用して、移動情報の交換を行なう必要がある。この場合における TCP コネクション再開手順は次のようになる。

1. 両端末はそれぞれ移動情報を移動情報管理サーバに送信する。
2. 移動情報管理サーバは、両端末から移動情報を受信すると、NAPT 内の端末に対して相手端末の移動先情報を送信する。
3. 2. で送信された移動先情報を受信した NAPT 内の端末は、相手端末の移動先に MSYN パケットを送信し、TCP コネクションの再開処理 (3.2 節参照) に移る。

3.4.4 両方の端末が NAPT 内にいる場合の TCP コネクション再開手順

両方の端末が NAPT 内にいる場合、3.3 節で説明した対称型プロキシサーバによって TCP コネクションを中継することによって、TCP コネクションを再開することができる。以下、その手順を示す。

1. 両端末はそれぞれ移動情報を移動情報管理サーバに送信する。
2. 移動情報管理サーバは、両端末から移動情報を受信すると、対称型プロキシサーバに中継対象とする TCP コネクションのリストをコネクション中継要求として送信する。
3. 対称型プロキシサーバは、移動情報管理サーバからコネクション中継要求を受信したとき、コネクションの数だけポートを確保して、そのポートのリストを移動情報管理サーバに返送する。
4. 移動情報管理サーバは、3. で返送されたポートのリストを受信すると、プロキシサーバの IP アドレスと割り当てを受けたポート番号を端末の移動先とした移動先情報を、両端末に送信する。

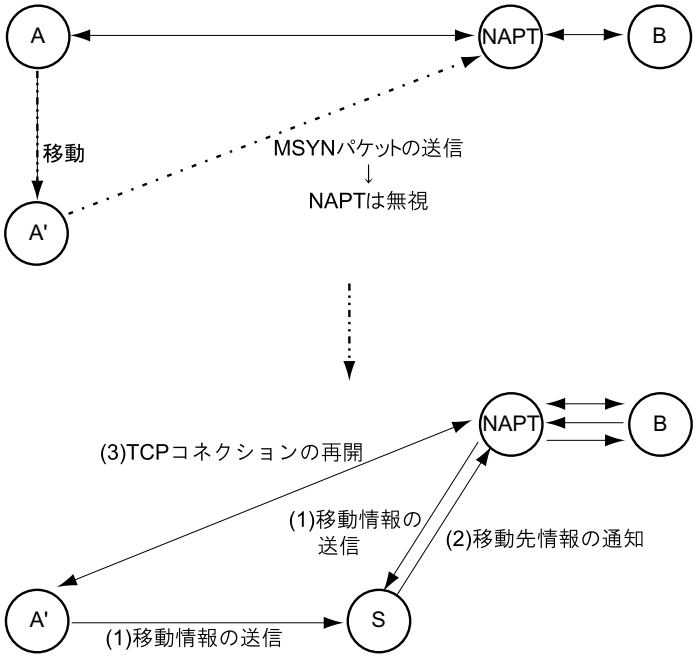

図 19: 片側に NAPT が存在する場合の TCP コネクション再開

5. 両端末は移動先情報を受信すると、プロキシが確保したポートに MSYN パケットを送信し、TCP コネクションの再開処理 (3.2 節参照) に移る。

3.4.5 プロキシ利用中の端末移動

3.4.4 節で示した手順によってコネクションが中継されているとき、端末の IP アドレスないしポート番号が変化した場合、次の手順で TCP コネクションを再開する。

1. 移動した端末は、3.2 節で示した手順に従って、直接の通信相手であるプロキシサーバに MSYN パケットを送信する。
2. MSYN パケットを送信したプロキシサーバは、Migrate-Permit オプションによって NAPT の存在を検出する。
3. 端末が外部に移動した場合は、プロキシサーバは相手端末に RST パケットを送信してコネクションの中継を中断する。その後、両端末は 3.4.3 節で示した手順に従って、TCP コネクションを再開する。
4. 端末が NAPT 内に移動した場合、プロキシサーバは端末に MSYN, ACK パケットを送信し、相手端末に Migrate-Permit オプションを付与したパケットを送信する。
5. 端末はこの SYN, ACK パケットに対する応答として、ACK パケットをプロキシサーバに送信する。
6. プロキシサーバは端末から ACK パケットを受信すると、TCP コネクションの中継を再開する。

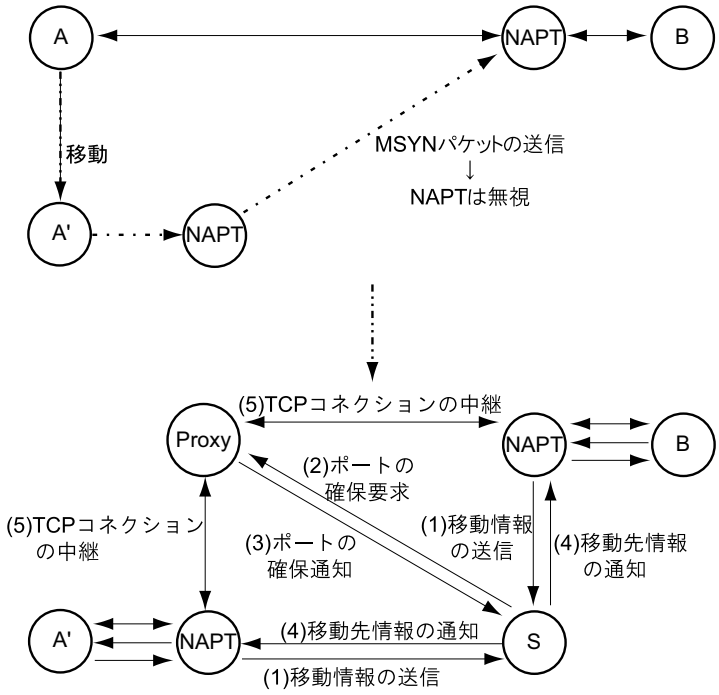

図 20: プロキシを利用した TCP コネクションの再開

プロキシによる中継において、最初の MSYN パケットと SYN, ACK パケットの交換で Migrate-Permit オプションを交換することによって、プロキシ中継を利用しない場合と同じように両端末は互いに相手端末の移動先ポートを知ることができる。また、プロキシサーバは TCP パケットのシーケンス番号の増加に関与しないため、シーケンス番号の調整と検証は通常の端末移動と同様に行なうことが可能である。

従って、プロキシによるコネクションの中継の後、端末移動によってプロキシを使用しない端末移動に切り替えた場合においても、通常の拡張 TCP によるコネクション再開が可能である。

4 実装と評価

第 3 章で提案した移動先情報交換プロトコルと拡張 TCP、TCP コネクション中継プロキシの動作と機能を評価するため、計算機上にこれらのプロトコルを実装し、実験を行なった。

4.1 実験環境

実験を行なうにあたって、移動端末として PC を 2 台、移動情報管理サーバと対称型プロキシとして PC を 1 台用意し、NAPT 内部への移動を検証するため、NAPT と DHCP の機能を有する小型ルータを 4 台用意した。いずれの PC も OS として Red Hat Linux 7.2 日本語版を使用した。

移動端末には拡張 TCP と移動情報管理クライアントを実装した。移動情報管理クライアントと移動情報管理サーバ、対称型プロキシは、Linux 上で動作するアプリケーションとして開発した。拡張 TCP は Linux 2.4.7 カーネルに対するパッチとして開発した。

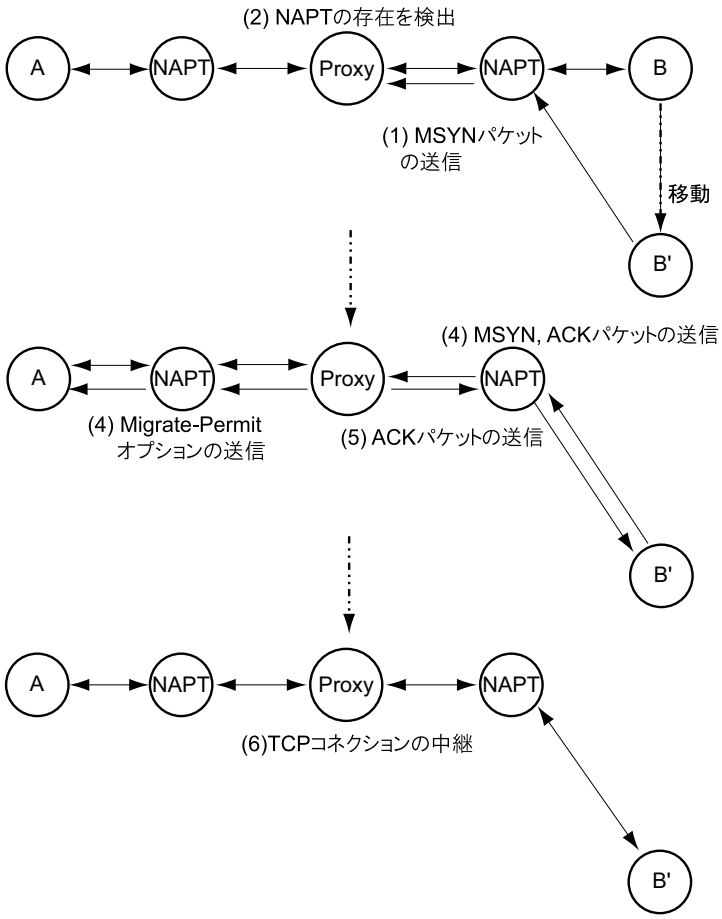

図 21: プロキシによる TCP コネクション中継の再開

実験は、3.4 節で示した各ネットワークトポロジを実際に構成して、端末の IP アドレスを変化させることによって、TCP コネクションが再開できるかどうかを確認する。また、移動端末において TCP パケットを監視することによって、3 ウェイハンドシェイクによる再開処理と TCP の再送制御が正常に動作しているかどうかを検証する。

4.2 評価

まず、図 22 に示すネットワーク構成において端末 B から端末 A に TCP コネクション経由でファイルを転送し、その途中で端末 A を別の NAPT 内へ移動させ、TCP コネクションの再開を行なった。端末 B においてパケットを観測した結果を図 23 に示す。ここで、hosts は端末 B のホスト名である。また、端末 B において観測しているため、端末 A の IP アドレスとして、NAPT の IP アドレスが記述されている。

図 23 のグラフの縦軸は ACK の確認番号、横軸は時刻を表す。

この結果では、端末 A の移動に約 8 秒かかっている。その後、NAPT(10.0.2.1) の配下に移動してから直ちに MSYN パケットを送信して TCP コネクションの再開を行なっている。端末 A の移動中にも端末 B は移動前の端末 A がいた NAPT(10.0.1.1) に対してパケットの再送を行なっている。

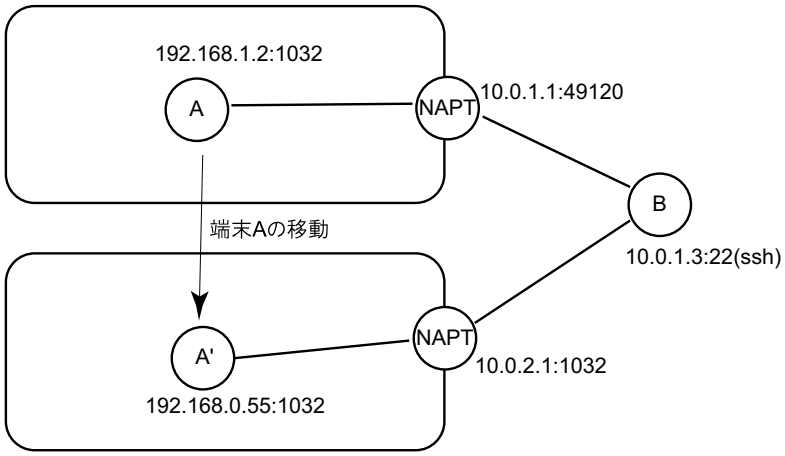

図 22: 実験環境

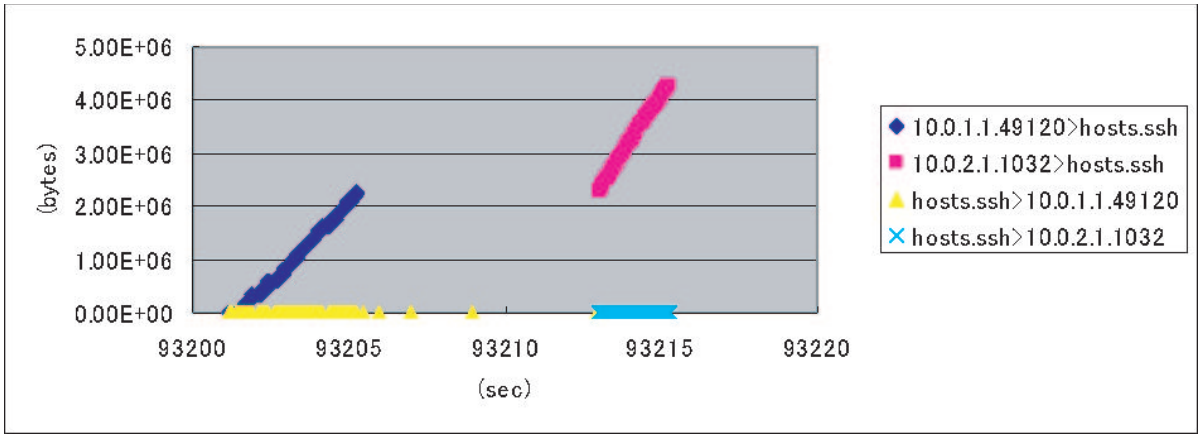

図 23: 端末移動時におけるパケットの観測結果

コネクションの再開に要した時間は、端末 B のパケット観測周期である 0.01 秒よりも短い時間であった。従って、MSYN パケットによる 3 ウェイハンドシェイクが完了してパケットの送信が再開されるまでの時間は非常に短いことが分かる。

5 おわりに

本研究では、互いに通信中の 2 つの端末が共に移動する場合、及び同時にそれぞれが NAPT 内部のネットワークへ移動する場合において、TCP コネクションを維持するためのプロトコルを提案した。

同時間帯における移動が発生した場合に相手端末の移動先 IP アドレスを知るための仕組みとして、移動先情報交換プロトコルを提案した。移動先情報交換プロトコルは、移動情報管理サーバが端末の移動先を相手端末に通知するために、サーバと端末間で移動情報を交換するプロトコルである。また、移動情報管理サーバは、端末が NAPT 内部のネットワークに移動したかどうかを検査

する役割も持つ。

通信を維持しようとする両方の端末が NAPT 内部に移動して、プロキシによる TCP コネクションの中継を行なうとき、移動情報管理サーバは両方の端末に対して、プロキシの IP アドレスとポート番号を相手端末の移動先として通知する。

端末が相手端末の移動先 IP アドレスとポート番号を移動先情報交換プロトコルによって知ることができたときに、TCP コネクションを再開できるようにするため、TCP を拡張した。TCP コネクションの再開は、コネクションの確立と同様の 3 ウェイハンドシェイクによって行なわれる。

通常のコネクション確立手順と区別し、移動の前後における端末の IP アドレスとポート番号を通知するため、Migrate-Request オプションを利用する。また、TCP コネクションの確立時には、相手端末が Migrate-Request オプションを解釈できるかどうかを確認し、かつ NAPT によって変換される前の IP アドレスとポート番号を相手ポートに通知するために、Migrate-Permit オプションを利用する。

Migrate-Request オプションを利用した TCP コネクションの再開手順によって、端末が移動する前に確立されていた TCP コネクションが、端末の移動先の IP アドレス及びポート番号において再開される。これによって、TCP コネクションを一旦解放することなく、アプリケーション間の通信を維持することが可能となる。

2 つの端末が両方 NAPT 内部のネットワークに移動した場合においても TCP コネクションを再開できるようにするため、対称型プロキシによる TCP コネクションの中継を提案した。対称型プロキシサーバは、両方の端末から TCP コネクションの確立要求を受信することによって、TCP コネクションの中継を開始する。

プロキシサーバの TCP コネクションの中継を利用している端末が移動した場合、移動先が NAPT 内部のネットワークであれば中継を維持し、そうでなければ中継を中断して、プロキシを介さずに相手端末と TCP コネクションを再開する手順に移る。

以上 3 つのプロトコルによって、互いに通信する両端末が同時間帯に移動しても、NAPT 内のネットワークに移動しても、移動先の IP アドレスを移動情報管理サーバが端末に通知し、拡張された TCP によって TCP コネクションが維持される。もし両端末が NAPT 内に移動した場合は、対称型プロキシによって TCP コネクションの中継を行なうことができる。

本研究では、NAPT による IP アドレス及びポート番号の変換機能を認識して、NAPT の存在の有無に応じて適切な手順で TCP コネクションの再開を行なうプロトコルを提案した。IP アドレス及びポート番号の変換に対処する仕組みであるため、NAPT による変換機能の対象となる UDP や ICMP(Echo 及び Echo Request メッセージのみ) 等に対しても、IP アドレスとポート番号の更新による通信の維持に関しては同様の議論が可能である。しかし、コネクションレス通信である UDP や ICMP は持続的に通信を行なうための仕組みを持たないため、TCP における SYN パケットの拡張のように端末の認証と通信の再開手順を新たに定義することは困難である。

本手法において、移動情報管理サーバ及び対称型プロキシサーバはどの端末からも接続可能としたが、実際のインターネット上での利用を考慮すると、過度の負荷を回避するために接続制限を設けるべきであると考えられる。しかし、互いに通信する両方の端末は同じサーバを参照しなければならない。従って、両方の端末に対して接続が許可されているサーバを利用するように、両端末が共通のサーバを選択する、もしくはサーバが端末を認証する仕組みを追加することが必要であると考えられる。

NAPT による IP アドレスとポート番号の変換を吸収するために、TCP オプションを利用して変換前後の IP アドレスとポート番号を交換する仕組みを設けた。しかし、実際には TCP オプションは全部で 40 バイト (オクテット) 以内で収まるようにしなければならない。他の一般的な TCP オ

プション (MSS[10] や SACK[11] など) を圧迫しないようにすることを考慮すると、多くの IP アドレス及びポート番号をそのまま TCP オプションとして付加することは実装上困難である。従って、本手法においてはコネクション ID という形でオプションの長さの節約を試みたが、本来は TCP オプションがより潤沢に利用できることが望ましい。

本手法の応用例として、端末間のプロセス移動 [12] が挙げられる。ネットワークで接続された複数の端末が協調して並列処理を行なう際、プロセスの負荷分散などの目的で、ある端末のプロセスを別の端末にネットワークを介して移動させることが考えられる。このとき、プロセスがそれまでに利用していた TCP コネクションを維持したままプロセスを移動させるためには、本手法による TCP コネクションの維持が役に立つものと考えられる。

参考文献

[1] C. E. Perkins, "IP Mobility Support," RFC 2002, October 1996.

[2] Xun Qu, Jeffrey Xu Yu and Richard P. Brent, "A Mobile TCP Socket," Joint Computer Science Technical Report Series of the Australian National University, April 1997.

[3] Alex C. Snoeren and Hari Balakrishnan, "An End-to-End Approach to Host Mobility," MIT Laboratory for Computer Science Cambridge, MA 02139, 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00), August 2000.

[4] P. Srisuresh, Jasmine Networks, K. Egevang, Intel Corporation, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, January 2001.

[5] Jon Postel, "Transmission Control Protocol," RFC 793, September 1981.

[6] M. Handley, ACIRI, H. Schulzrinne, Columbia U, E. Schooler, Cal Tech, J. Rosenberg, Bell Labs, "SIP: Session Initiation Protocol," RFC 2543, March 1999.

[7] "Session Initiation Protocol (SIP)," http://www.cs.columbia.edu/~hgs/sip/

[8] 井上 淳, "Mobile IP 概要," Internet Week 99 チュートリアル講演資料, http://www.soi.wide.ad.jp/soi/iw99/iw99_tut/slides/14/ , December 1999.

[9] W. Richard Steven, 篠田 陽一, "UNIX ネットワークプログラミング 第 2 版 Vol.1," ピアソン・エデュケーション, July 1999.

[10] V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance," RFC 1323, May 1992

[11] Mathis, et. al, "TCP Selective Acknowledgement Options," RFC 2018, October 1996.

[12] Marian Bubak, Dariusz bik, Dick van Albada, Kamil Iskra, Peter Sloot, "Portable Library of Migratable Sockets ," Proceedings of the SGI Users' Conference 2000, October 2000.