

NTMobileにおける 最適なりレーサーバ選択手法の提案と実装

三宅 佑佳¹ 鈴木 秀和¹ 内藤 克浩² 渡邊 晃¹

概要: スマートフォン等の移動通信端末の普及、及び無線通信技術の発展により、通信接続性と移動透過性が要求されている。通信接続性とは、ネットワーク環境によらず確実に通信が開始できることであり、移動透過性とは、通信中にネットワークを切り替えても継続して通信できることである。NTMobile (Network Traversal with Mobility) はこれらの機能を同時に実現できる次世代の技術である。NTMobile では基本的に端末間で直接通信を行うが、相手通信端末が一般サーバであるなど、直接通信ができない場合はRS (Relay Server) を経由した通信を行う。しかし、RS を経由する場合、直接通信を行う場合と比べて通信経路が冗長になる。本論文ではNTMobile において最適なRS を選択し、通信経路の冗長を抑制する手法を提案する。また、Linux 上で提案方式のプロトタイプを実装し、仮想環境上で動作検証を行った。プロトタイプの性能評価を行い、RS 選択にかかるオーバーヘッドは無視できる程度であることを確認した。

1. はじめに

スマートフォンのような移動通信端末の普及や無線通信技術の発展により、ネットワーク利用の需要が急激に増加している。それに伴い、ネットワークを切り替えても継続して通信を行いたいという要求が高まっている。

IPv4 ネットワークでは、グローバルアドレスの枯渇が深刻化している。NAT (Network Address Translation) は、配下の通信端末にプライベートアドレスを割り当て、プライベートアドレスとグローバルアドレスを変換することにより、グローバルアドレスの消費を回避する。しかし、NAT を用いたネットワーク構成は、グローバルネットワーク側からNAT 配下のプライベートネットワークに対して通信を開始できないため (NAT 越え問題)、通信端末の双方向通信を妨げる要因となる。一方、長期的な解決策として、IPv6 ネットワークの導入が準備されているが、IPv6 アドレスはIPv4 アドレスとの互換性が無いため、普及が滞っている。そのため、今後もIPv4 アドレスとIPv6 アドレスが混在した環境が長らく続くことが想定される。このような背景から、接続しているネットワークの構成に関わらず自由に通信を開始可能な通信接続性の実現が期待されている。

IP ネットワークでは、各通信端末に割り当てられたIP アドレスにより通信が識別されている。IP アドレスは位置情報を持つため、ネットワークが切り替わると変化し、通信を継続することができない。そこで、ネットワークを切り替えても通信を継続可能な移動透過性の実現が要求されている。

移動透過性を実現する技術として、MIPv4 (Mobile IPv4) [1], MIPv6 (Mobile IPv6) [2], DSMIP (Dual Stack Mobile IPv6) [3] が標準化されている。これらの技術は、アドレス管理機能とパケット中継機能を持つHA (Home Agent) を用いて通信を行い、NAT 越えや一般サーバとの通信も実現できる。しかし、HA を経由する通信は経路が冗長になるという課題がある。

筆者らは、通信接続性と移動透過性をIPv4/IPv6 混在環境において実現するNTMobile (Network Traversal with Mobility) を提案している [4], [5], [6], [7]。NTMobile は基本的に端末間で直接通信を行うが、直接通信を行うことができない環境下においては通信中継の役割を担うRS (Relay Server) を経由した通信となる。RS はグローバルネットワーク上に分散配置が可能であり、複数のRS から通信に使用するRS を自由に選択することができる。そこで、本論文では通信端末とRS との間のルータ経由数を調査し、最適なRS を選択することによって通信経路の冗長を抑制する手法を提案する。また、提案手法のプロトタイプを実装し、動作検証、及び性能評価を行う。

以後、2 章では既存技術について、3 章ではNTMobile

¹ 名城大学大学院 理工学研究科
Graduate School of Science and Technology, Meijo University

² 愛知工業大学 情報科学部
Information Science, Aichi Institute of Technology

について述べる。4章では提案するRSの選択手法について説明し、5章では提案手法の実装と動作検証の結果、性能評価を示す。最後に6章でまとめる。

2. 既存技術

2.1 Mobile IPv4

MIPv4[1]は、IPv4ネットワークを対象とした移動透過技術である。MIPv4では、移動により変化しないアドレスHoA(Home Address)と、位置識別子として移動先ネットワークで割り当てられるCoA(Care of Address)の2種類のアドレスを持つ。アプリケーションがHoAを用いて通信を行うことにより、端末が移動した際にCoAの変化を隠蔽し、移動透過性が実現されている。

MNは起動時に利用するHAを決定し、そのHAからHoAが割り当てられる。通信相手端末CN(Correspondent Node)はMNの位置に関わらず、MNのHoAを宛先にして通信を行う。MNが異なるネットワークに移動すると、移動先のネットワークでCoAを取得し、HAにHoAとCoAを対応付けて登録する。HAはホームネットワークにおいて、MNのHoA宛てに送信されたパケットを受信すると、MNが移動した先のネットワークに送信する。このとき、パケット内の宛先アドレスがHoAであり、移動先のアドレス帯域と異なるため、MNとHAとの間にトンネルを構築してパケットのカプセル化を行う必要がある。MNはHoA宛てのパケットを受信すると、送信元をHoAとしたCN宛てのパケットを送信する。このようにしてMNとCNは通信を継続することができる。

しかし、MNからCNまでの経路上のルータがインGRESSフィルタリングを行っている場合、MNが送信元IPアドレスを偽造しているものと判断され、パケットが破棄される恐れがある。そのため、MNからCN宛のパケットもHAを経由するReverse Tunneling[8]という方法が定義されている。

MIPv4はNAT越えを実現するために仕様が拡張されている[9]。図1にNAT越えを実現したMobile IPv4を示す。MNがNAT配下のネットワークに移動すると、MNとHAの間でトンネルを構築することでNAT越えが実現される。すなわち、NAT環境ではReverse Tunnelingが必ず適用される。また、MNは立ち上げ時にHAを動的に選択ができるよう拡張されている[10]。

しかし、MIPv4には多くの課題が存在する。MIPv4では、HAを常に経由した通信になるため、通信経路が冗長になり、スループットの低下やネットワーク負荷増大の懸念が課題として挙げられる。MNは起動時にHAを選択することができるが、その後は変更できないため、MNが移動した際に通信経路冗長化が懸念される。HAの設置場所は必ずグローバル上に設置されるため、MNごとにHoAとしてグローバルアドレスが必要になる。これは、IPv4ア

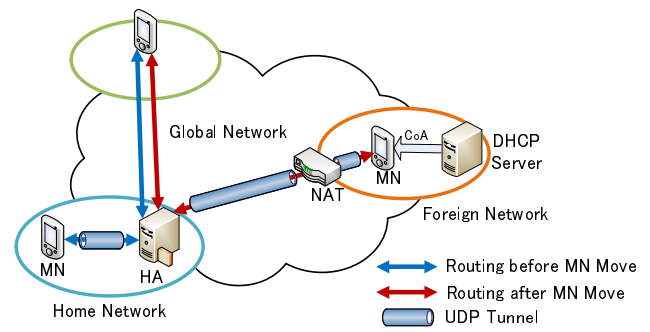


図1 Mobile IPv4のネットワーク構成

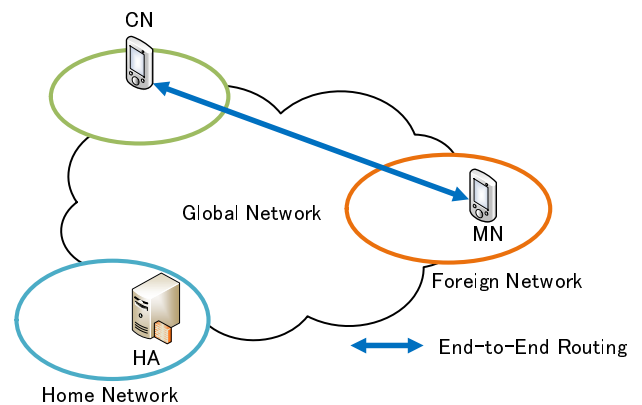


図2 Mobile IPv6のネットワーク構成

ドレス枯渇問題に逆行する課題である。

2.2 Mobile IPv6

MIPv6はIPv6ネットワークを対象とする移動透過技術である。図2にMIPv6のネットワーク構成を示す。MIPv6では、ネットワークとアプリケーションがIPv6である必要がある。

MIPv6では、エニーキャスト[11]を用いたDynamic Home Agent Address DiscoveryというHAを選択できる方式が定義されている[12]。エニーキャストを用いると、複数あるノードのうち最も近い位置にある宛先にパケットを送信するため、通信経路冗長化の抑制や負荷分散に利用される。また、MIPv6では経路最適化により通信端末同士の直接通信が可能である[13]。

しかし、MIPv6では経路最適化により、通信端末間で直接通信が可能であるが、通信相手がMIPv6の機能を持たない一般サーバGS(General Server)との通信の場合はHA経由の通信となる。MIPv6では、MNが起動時にHAを選択することができるが、その後は変更できないため、MNが移動した際に通信経路冗長化が懸念される。

2.3 Dual Stack Mobile IPv6

図3にDSMIPのネットワーク構成を示す。DSMIPでは、MNがIPv4ネットワークに移動したとき、移動先ネッ

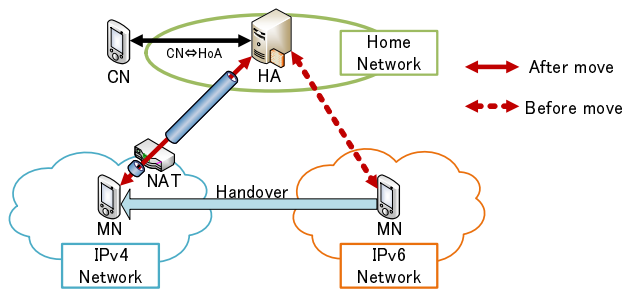


図 3 Dual Stack Mobile IPv6 のネットワーク構成

トワークで取得した CoA (IPv4) を MN の HoA (IPv6) と紐付け、IPv4 ネットワークからも通信が可能となる技術である。DSMIP の HA はデュアルスタックネットワーク上に設置する必要がある。また、DSMIP は、MIPv6 を拡張した技術であり、アプリケーションは IPv6 である必要がある。

3. NTMobile

3.1 NTMobile の概要

NTMobile は、仮想 IP アドレスに基づいた通信をアプリケーションが行うことにより、移動透過性を実現する技術である [4], [5], [6]。仮想 IP アドレスは、ネットワークの切り替えにより変化せず、実ネットワークに依存しないアドレスである。そのため、通信中にネットワークを切り替えた場合でも、IP アドレスの変化をアプリケーションに対して隠蔽し、通信を継続することができる。また、アプリケーションは IPv4, IPv6 どちらの場合でも通信可能である。

図 4 に NTMobile のネットワーク構成を示す。NTMobile は NTMobile を実装した NTM 端末、通信中継装置 RS、NTM 端末や RS のアドレス情報を管理する DC (Direction Coordinator) によって構成される。DC は、NTM 端末に対して仮想 IP アドレスを割り当てる役割を持つほか、NTM 端末と RS にトンネル構築の指示を行う役割を担う。DC 及び RS は、グローバルネットワーク上に分散配置することが可能である。また、RS は中継装置として独立しており、通信ごとに適切な RS を選択することができる。

NTM 端末は、起動時に自身の実 IP アドレス等のアドレス情報を DC に対して登録する。その際、NTM 端末は DC から仮想 IP アドレスを割り当てられ、NTM 端末のアプリケーションは仮想 IP アドレスを端末識別子として利用する。また、全ての packets は実 IP アドレスによってカプセル化される。すなわち、NTM 端末の実 IP アドレスは位置識別子としてパケットのルーティングに利用される。

3.2 RS の役割

NTMobile では、基本的に NTM 端末間で直接通信を行うが、直接通信ができない場合は RS を経由した通信となる。

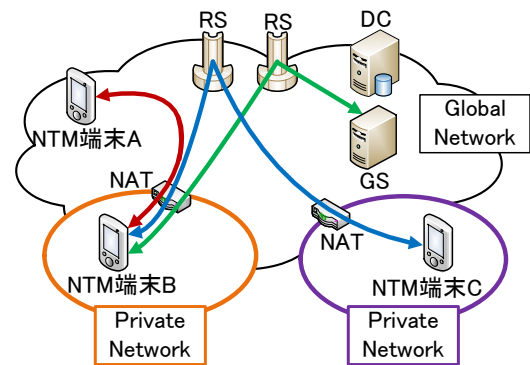


図 4 NTMobile のネットワーク構成

直接通信できない通信環境とは、異なる NAT 配下に存在する NTM 端末同士の通信の場合、NTM 端末と NTMobile を実装していない GS との通信の場合、更に、IPv4/IPv6 混在環境での相互通信の場合である。RS が通信を中継することによって、端末間で直接通信ができない環境下でも接続性を保障することができる。また、NAT 配下の NTM 端末同士の通信においては、自律的経路最適化を適用することによって、MN と CN の間で直接トンネルを構築して通信を行うことができる場合がある [14]。ただし、MN 側と CN 側の両 NAT が Symmetric NAT であると、経路最適化が適用されないため、RS を経由した通信となる。

3.3 NTMobile の動作

3.3.1 NTM 端末のアドレス情報登録処理

NTM 端末は起動時とネットワーク切り替え時に、DC に対してアドレス情報の登録を行う。NTM 端末は、NTM 端末の実 IP アドレスや FQDN 等の情報を載せた NTM Registration Request を DC に対して送信する。DC は NTM 端末の FQDN から NTM 端末が一意に決まる Node ID を生成する。また、DC のデータベースに受信した NTM 端末の端末情報を登録し、NTM 端末に仮想 IP アドレスを割り当てる。そして、DC は NTM 端末に対して仮想 IP アドレス等を記載した NTM Registration Response を返信する。

3.3.2 NTM 端末同士の通信におけるトンネル構築処理

図 5 に MN から CN に対して通信を開始する際のトンネル構築シーケンスを示す。図 5 は、MN と CN は異なる NAT 配下に存在する場合の例である。はじめに、MN は DC_{MN} に対して CN の名前解決及びトンネル構築の依頼をするため、NTM Direction Request を送信する。NTM Direction Request には、CN の FQDN (FQDN_{CN}) が記載されている。DC_{MN} は MN からの依頼を受けると、DNS Request / Response for NS Record によって、CN の NS レコードを取得する。そして、通信相手が NTM 端末であるか否かを判断するため、CN を管理している DC_{CN} に対して、DNS Request for TXT Record を送信し、TXT レコード

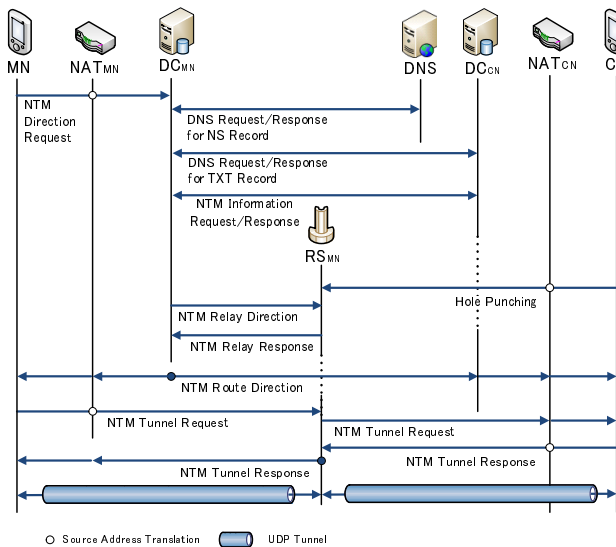


図 5 NTM 端末同士のトンネル構築シーケンス

の問い合わせを行う。DCにはTXTレコードに自身がDCであることを示すTXTレコードが登録されているため、DCMNはDCCNからDNS Request for TXT Recordが返ると、CNがNTM端末であることが判明する。DCMNは、FQDNCNを記載したNTM Information Requestを送信し、それを受信したDCCNはNTM Information ResponseにCNの端末情報を載せ、DCMNへ応答を返す。DCMNはMN及びCNのアドレス情報より、ネットワーク上の位置を把握する。この場合、MNとCNは異なるNAT配下に存在するため、DCMNはRSを経由して通信を行うことを決定する。DCMNは自身の管理下にあるRSMNに対し、MNとCNの間にトンネル構築を行うよう指示するため、NTM Relay Directionを送信する。NTM Relay Directionを受信したRSMNは、トンネル構築が可能であることをDCMNに伝えるため、NTM Relay ResponseをDCMNに送信する。DCMNはNTM Route Directionを、MNに対しては直接、CNに対してはDCCNを経由して送信する。指示を受けたMNは、RSMNとトンネル構築するため、NTM Tunnel RequestをRSMNを経由してCNに送信する。CNはRSMNを経由してMNに対してNTM Tunnel Responseを返す。この動作により、MNとCN間にRSMNを経由したトンネルが構築される。

3.3.3 NTM 端末と一般サーバの通信におけるトンネル構築処理

NTM 端末の移動透過性を確保するため、GS との通信を中継するRSがパケットのカプセル化・デカプセル化、及び仮想IPアドレスと実IPアドレスのアドレス変換を行う。GSは通信相手をRSとして認識する。

図 6 に MN から GS に対して通信を開始する際のトンネル構築シーケンスを示す。まず、MN は DCMN に対して GS の名前解決及びトンネル構築依頼のため、NTM Di-

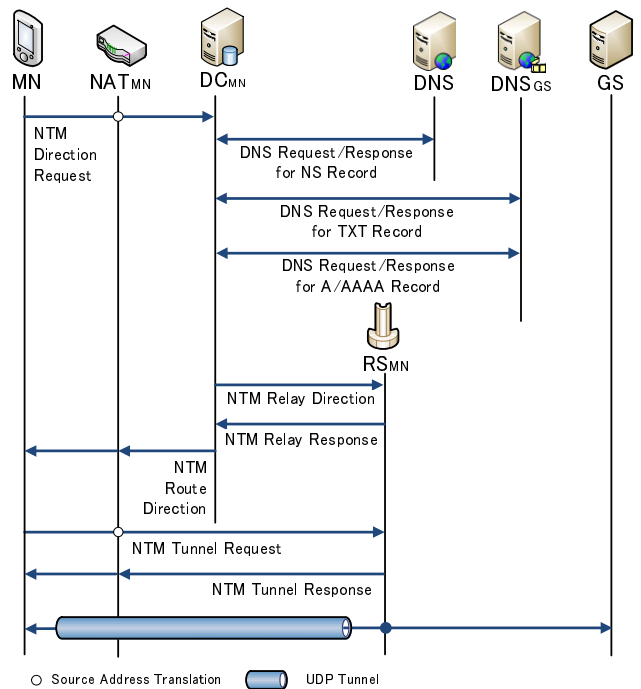


図 6 NTM 端末と一般サーバとの通信シーケンス

rection Request を送信する。依頼を受けた DCMN は DNS Request / Response for NS Record により、GS の権威サーバ DNSGs のアドレスを取得する。次に、DNSGs に対して DNS Request/Response for TXT Record の交換を行い、通信相手端末が NTMobile の機能を持たない GS であると判断する。DCMN は通常の DNS Request / Response for A / AAAA Record により、GS のアドレス情報を取得する。DCMN は GS のアドレス情報を記載した NTM Relay Direction を RSMN に送信し、MN との間で通信の中継を行うよう指示する。RSMN はアドレス変換テーブルを生成後、NTM Relay Response を返信する。その後、DCMN は MN に対して RSMN との間にトンネル構築するよう指示するため、NTM Route Direction を送信する。その指示を受け取った MN は RSMN に対して NTM Tunnel Request/Response を交換し、RSMN との間においてトンネル構築を行う。

3.4 RS を経由する通信経路の課題

RS を経由する通信において、通信開始時に適切な RS が選択されないと、NTM 端末と通信相手端末間の通信経路が冗長になる可能性がある。特に、NTM 端末と GS との通信においては、通信中に RS を切り替えることができないため、NTM 端末の移動後を考慮せず RS を選択した場合、更に経路が冗長になることが懸念される。

通信時に冗長な経路をとることによる課題として、パケット伝送遅延の増加、スループットの低下、更に、ネットワーク負荷の増大が挙げられる。そのため、通信時において通信経路が冗長にならず、NTM 端末がネットワーク

切り替え後についても考慮した RS を選択する手法を確立する必要がある。

4. 提案方式

4.1 提案の概要

NTM Mobile において中継機能は RS として独立しており、通信ごとに RS を選択することができる。また、NTM 端末同士の通信においては、通信中に経路を切り替えた際、RS も切り替えることができる。そこで、通信経路の冗長化を抑制するため、経路構築時に通信経路のルータ経由数（ホップ数）が最少となるような RS 選択手法を提案する。NTM 端末同士の通信の場合、NTM 端末が起動した際、それぞれ NTM 端末から RS までのホップ数調査を行う。その結果を基にし、それぞれの NTM 端末から各 RS までのホップ数を算出し、その中から最少ホップ数となる最適な RS を選択する。NTM 端末と GS との通信の場合は、通信開始時に RS と GS までのホップ数調査を行う。GS との通信の場合、NTM 端末は移動する可能性があるが、GS は移動を行わない。また、GS は RS を通信相手と認識して通信を行うため、通信中に RS を切り替えることができないという制約がある。そのため、DC は各 RS と GS 間のホップ数を比較し、ホップ数が最少となる RS を選択する。

4.2 RS の評価指標

RS の評価は、通信経路上のルータのホップ数とする。IPv4 ネットワークにおけるホップ数は IP ヘッダ内の TTL (Time to Live) を用いて調査する。TTL は IP パケットがルータを経由するごとに値が減少するため、初期値との差を算出することでホップ数を取得する。

通信経路の評価指標として、パケットの往復遅延を示す RTT (Round Trip Time) を利用する方法が考えられる。しかし、NTM 端末が接続するネットワークは無線環境であるため、帯域の狭いネットワークは RTT が比較的長く、振れ幅が大きい。そのため、NTM 端末から RS までの RTT を正確に測定するには、多数の制御メッセージの往復が必要である。そのため、ネットワークと端末にかかる負荷と、ネットワーク接続時のオーバーヘッドが増大し、NTM 端末が移動を行うほどその影響が大きくなる。

図 7 に Google Public DNS (IP アドレス: 8.8.8.8) 宛てに、自宅のパソコン (有線環境と無線環境)、大学のパソコン (有線環境と無線環境) からそれぞれコマンド `tracert` を実行して RTT を計測した結果を示す。実験日は 2015 年 7 月 23 日から 2015 年 7 月 30 日までの 8 日間であり、それぞれの環境において 1 日 5 回ずつ `tracert` を実行した結果である。図 7 において、同じ環境下で有線での通信と無線での通信を比較すると、自宅環境においても大学環境においても無線通信の方が揺らぎが大きくなり、通信速度が遅くなる。また、RTT について、プロバイダ内を通過すると

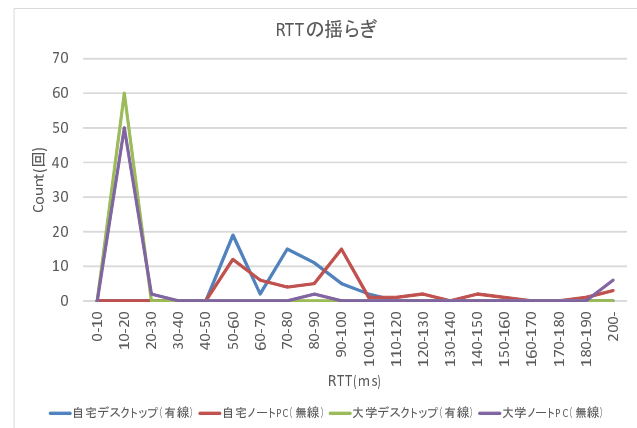


図 7 RTT の計測結果

きは小さくなり、太平洋を跨ぐときは大きくなり、通信環境に左右される。これらのことから総合的に判断し、RTT により最適経路を発見することは難しいと考える。一方、ホップ数は通信経路の環境に依存する指標であるため、接続したネットワークごとに制御メッセージを 1 回送信するだけでよい。

CAIDA (The Cooperative Association for Internet Data Analysis) [15] による RTT とホップ数の関連性の調査により、ホップ数が増加すると、それに伴い RTT も上昇する傾向があることがわかっている。従って、通信経路のホップ数を少なくすることで伝送遅延を抑え、スループットを向上させることができると考えられる。以上の理由により、ホップ数を評価指標として採用した。

4.3 ホップ数調査の方法

NTM 端末同士の通信を行う場合と、NTM 端末と GS との間で通信を行う場合と分けて説明する。これまでの提案シーケンスでは、RS が踏み台攻撃の対象になってしまうという懸念や、前者と後者のホップ数調査シーケンスの統一性がないという問題があった。それを解消するため、本論文では前者と後者両方のホップ数調査において、ICMP を利用する手法を提案する。両者の違いは、前者では両 NTM 端末がネットワークを切り替えたタイミングでホップ数調査を行うのに対し、後者では NTM 端末と GS の通信開始時にホップ数調査を行う。また、前者では NTM 端末と RS 間で調査を行い、それを基に最適な RS を選択するが、後者では GS と RS 間でホップ数調査を行うという違いがある。

4.3.1 NTM 端末同士の通信におけるホップ数調査

図 8 に NTM 端末から RS までのホップ数調査のシーケンスを示す。NTM 端末は自身の起動時、もしくはネットワーク切り替え後に、DC に対してアドレス情報の登録処理を行う。DC は NTM 端末のアドレス情報登録処理を行うと、NTM 端末に対して、調査対象となる RS の IP アド

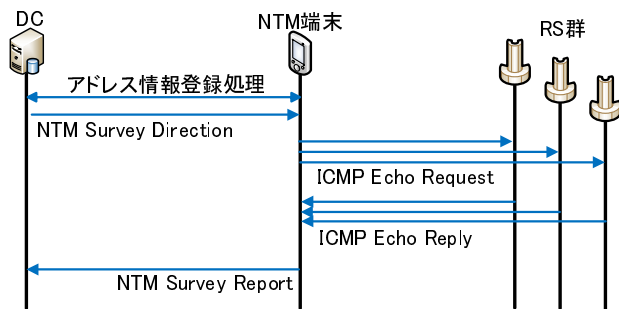


図 8 NTM 端末同士の通信時におけるホップ数調査シーケンス

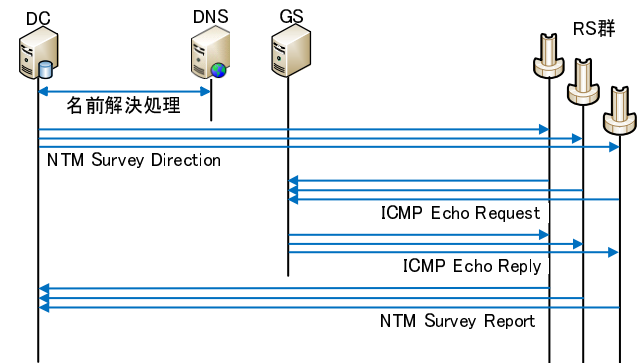


図 9 NTM 端末と一般サーバとの通信シーケンス

レスなどを記載した NTM Survey Direction を送信し、各 RS までのホップ数調査を行うよう指示する。NTM Survey Direction を受信した NTM 端末は、各 RS に対してホップ数調査を行うため、ICMP Echo Request を送信し、その応答である ICMP Echo Reply を受信する。NTM 端末は ICMP Echo Reply の IP ヘッダにある TTL の値と TTL の初期値との差分をとり、ホップ数を算出する。TTL の初期値は OS (Operating System) によって異なるが、NTM 端末が受け取った TTL の値から推測が可能であり、ホップ数を算出できる。このとき、タイムアウトを設け、ホップ数調査対象となる RS の中から一定の時間内に ICMP Echo Reply が返ってきた RS のみホップ数を算出する。NTM 端末はホップ数を算出すると、NTM Survey Report に、RS の IP アドレス、NTM 端末から RS までのホップ数を記載し、調査指示を出した DC に対してホップ数調査の結果を報告する。DC は NTM Survey Report を受信すると、NTM 端末から RS までのホップ数を Hop Table に記録する。

4.3.2 NTM 端末と一般サーバとの通信におけるホップ数調査

図 9 に RS から GS までのホップ数調査のシーケンスを示す。GS との通信開始時、DC は GS の名前解決を行うため、一般 DNS サーバとの間で名前解決処理を行う。そして、DC が GS の A/AAAA レコードを取得すると、GS と DC 管理下の全ての RS との間のホップ数調査を開始する。

DC は管理下の RS に対して、GS の IP アドレスを載せた NTM Survey Direction を送信し、各 RS までのホップ数調査を行うよう指示する。NTM Survey Direction を受信した各 RS は、GS に対してそれぞれ ICMP Echo Request を送信する。各 RS は、GS から ICMP Echo Reply が返ってくると、そのパケットの IP ヘッダにある TTL の値と TTL の初期値との差分をとり、ホップ数を算出する。このとき、タイムアウトを設け、ホップ数調査対象となる RS の中から一定の時間内に ICMP Echo Reply が返ってきた RS のみホップ数を算出する。その後、RS は調査指示を出した DC に対してホップ数調査報告をするため、NTM Survey Report を送信する。DC は NTM Survey Report を受信す

ると、NTM 端末から管理下の RS までのホップ数を Hop Table に記録する。

4.4 RS の選択

通信経路冗長化を抑制する最適な RS の選択について、NTM 端末同士の通信の場合、NTM 端末と GS との通信の場合に分けて説明する。

図 10 に NTM 端末同士の RS を経由した通信を開始するシーケンスを示す。この場合、MN と CN は DCMN-CN 管理下の全ての RS に対して既にホップ数調査を行っており、DCMN-CN の Hop Table に図 10 のような調査結果が登録されている。

DCMN-CN は MN より NTM Direction Request を受け取ると、DCMN-CN の Hop Table の中から最少ホップ数となる最適な RS を選択する。DCMN-CN は Hop Table において、MN から各 RS までのホップ数、及び CN から各 RS までのホップ数の情報を、MN や CN の Node ID、RS の IP アドレスをキーとして検索する。そして、MN から各 RS までのホップ数と CN から各 RS までのホップ数を合算し、総経路ホップ数を算出する。図 10 では DCMN-CN は算出した各総経路ホップ数の中から、ホップ数が最少の 25 となる RSx を選択し、トンネル構築までの経路指示手順を実施する。

図 11 に MN と GS との通信時における RS 選択のシーケンスを示す。NTM 端末と GS が通信を行う場合、DCMN が GS の A/AAAA Record を取得すると、図 9 に従ってホップ数調査が開始されるが、図 11 ではホップ数調査のシーケンスは省略する。ホップ数調査後、DCMN は Hop Table の中から最もホップ数の少ない RSA を選択し、トンネル構築までの経路指示手順を実施する。ここで、GS によっては、ICMP パケットが通らないネットワークに設置されている場合が考えられる。この場合には、MN と RS 間で最少ホップ数となる RS を選択する。

5. 実装と評価

本章では、提案方式の実装とその動作検証、及び性能評

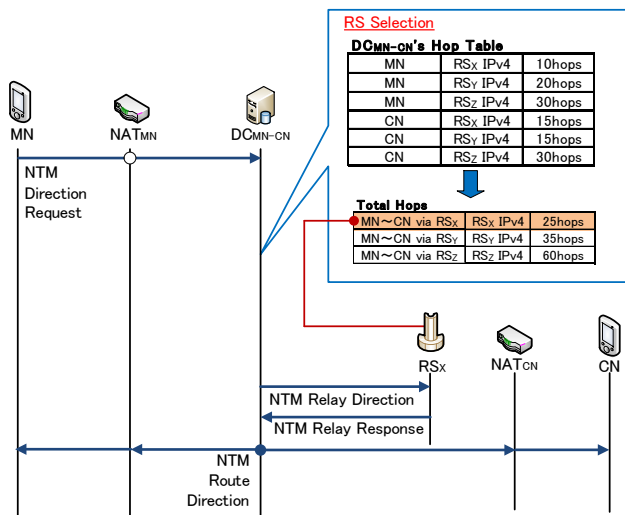


図 10 NTM 端末同士の通信時における RS 選択

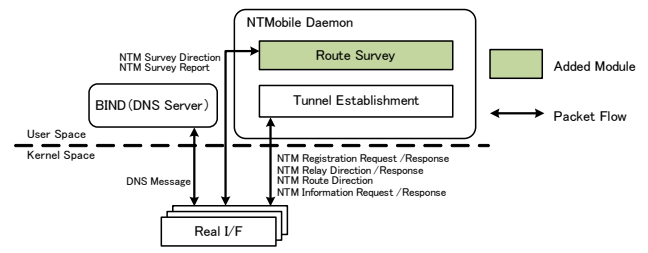


図 12 DC のモジュール構成

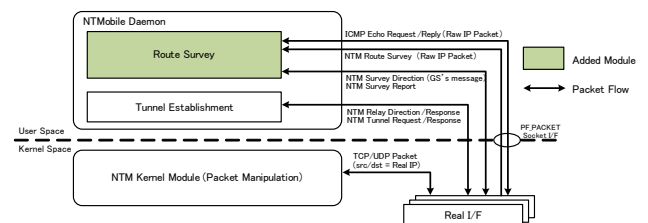


図 13 RS のモジュール構成

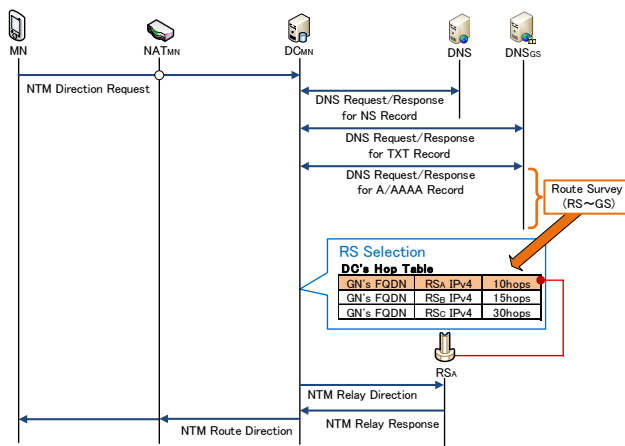


図 11 NTM 端末と一般サーバとの通信時における RS 選択

価について述べる。

NTMobile の基本動作は Linux 環境での実装が行われており、動作が確認されている。提案方式である RS と GS 間のホップ数調査についてのプロトタイプを、DC, RS へそれぞれ追加実装を行った。

動作検証として、提案するホップ数調査の動作が正常に行われるかどうか確認した。また、提案方式の評価として、ホップ数調査を行うことにより生じる処理遅延を測定した。

5.1 実装

提案方式のプロトタイプの実装構成を以下に示す。

5.1.1 DC への実装

図 12 に DC のモジュール構成を示す。DC はユーザ空間の制御メッセージの送受信を行う NTM デーモンと、BIND を利用した DNS サーバにより構成される。DC の NTM デーモンに、ホップ数調査を行うモジュールとして Route Survey のプロトタイプの実装を行った。

NTM 端末同士の通信の場合は、DC が NTM 端末のアドレス情報の登録が完了したタイミング、NTM 端末と GS

との通信の場合は、DC が GS のアドレス情報を取得したタイミングでホップ数調査を開始するようプロトタイプを作成した。

NTM 端末同士の通信の場合における Route Survey では、MN から RS まで、及び CN から RS までの経路調査の結果から、ホップ数が最少となる最適な RS が選択できるよう処理を記述した。また、NTM 端末と GS の通信における Route Survey には、GS と RS までのホップ数調査結果を基に最適な RS 選択ができるよう処理を記述した。

5.1.2 RS への実装

図 13 に RS のモジュール構成を示す。RS はユーザ空間の NTM デーモンと、カーネル空間の NTM カーネルモジュールによって構成される。RS には NTM デーモンに、ホップ数調査を行うモジュールとして Route Survey のプロトタイプの実装を行った。

NTM 端末と RS の間の経路調査を行う場合は Route Survey の IP ヘッダから TTL を取得する必要がある。そのため、RS ではデバイスレベルのパケットインターフェースである PF_PACKET を利用することによって、Route Survey が IP ヘッダを含むパケットを受信可能とした。

GS と RS の間の経路調査を行う場合は ICMP Echo Request / Reply を Raw socket により、送受信を行うことによって、IP ヘッダから TTL を取得することを可能とした。

5.2 動作検証

表 1 にホスト PC の構成、表 2 に仮想マシンの構成、図 14 に動作検証におけるネットワーク構成を示す。1 台のホスト PC 上に VMware Player^{*1} を用いて、DC, MN, 3 台の RS (RSA~RSC), DNS サーバ, GS, ルータを構築した。DC, MN, RSA, DNS サーバ, GS を同一 IPv4

*1 <http://www.vmware.com/jp/>

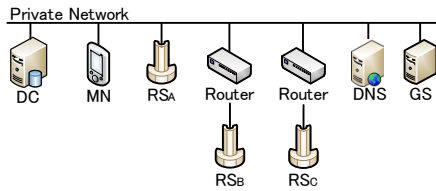


図 14 動作検証におけるネットワーク構成

表 1 ホスト PC の構成

| ホスト PC | |
|--------|----------------------------|
| OS | Windows7 64bit |
| CPU | Intel Core i7-2600 3.40GHz |
| Memory | 8.00GB |

ネットワークに接続し、このネットワークと RS_B 及び RS_C との間に 1 つルータを接続した。この動作環境において、ホップ数調査が正常に行われることを確認した。

5.3 性能評価

GS と RS 間のホップ数調査における処理時間を計測し、性能評価を行った。図 14 の環境において実行した GS と RS 間のホップ数調査実施時間を図 15 に示す。ホップ数調査にかかった時間は DC にて Wireshark^{*2} により取得した。図 15 について、GS と NTM 端末から GS に対して通信を 25 回行った平均時間を示す。

5.3.1 仮想環境におけるホップ数調査

図 15 は、DC が DNS サーバから DNS Response for A/AAAA Record を受信してから、各 RS からの NTM Survey Report を受信するまでの時間の内訳を示す。DC が、DNS サーバから DNS Response for A/AAAA Record を受信してから、RS_A に対する NTM Survey Direction を送信するまでにかかった時間は 1.584ms であった。その後、送信された RS_B に対する NTM Survey Direction は 0.915ms、RS_C に対する NTM Survey Direction は 0.695ms であった。DC において、DNS Response for A/AAAA Record を受信してから NTM Survey Direction を送信するまでの間には、ホップ数調査用のスレッドを生成する処理があり、それによる遅延が発生していると考えられる。

RS_A, RS_B, RS_C から GS に対して ICMP Echo Request を送信し、GS が各 RS に ICMP Echo Reply を返すまでの時間は、それぞれ 1.408ms, 3.598ms, 3.776ms である。RS_A は、RS_B, RS_C と比べてメッセージの送受信に時間がかかっているが、これはルータを経由しているためである。また、RS_A, RS_B, RS_C が ICMP Echo Request を GS から受け取り、DC に対して NTM Survey Report を送信するまでにかかる時間は、1.996ms, 2.248ms, 2.521ms である。この時間には、各 RS において GS までの間のホップ数を算出する処理の遅延が含まれている。

*2 <https://www.wireshark.org/>

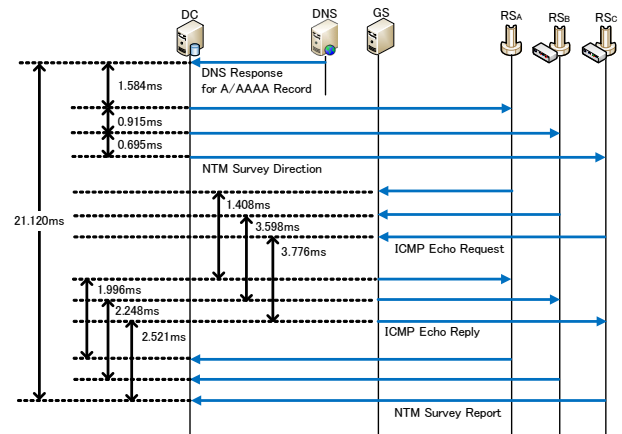


図 15 GS と RS 間のホップ数調査時間

この環境において、GS の名前解決処理をトリガーにしたホップ数調査は 21.120ms で完了した。

5.3.2 実環境におけるホップ数調査

実環境での調査時間の検討のため、NTM 端末、DC、各 RS、DNS サーバ、GS が日本国内のグローバルネットワーク上に存在し、NTM 端末が 3G ネットワークに接続することを想定する。また、[6]の結果より、グローバルネットワークの RTT を約 20ms, 3G ネットワークの RTT を約 120ms と仮定した。

GS から RS までのホップ数調査のシーケンスにおいて、NTM Survey Direction, ICMP Echo Request, ICMP Echo Response, NTM Survey Report はそれぞれ送信時に 10ms 程度の一方向遅延が生じる。従って、図 15 の場合における GS から RS までのホップ数調査は、61.12ms 程度で正常に完了すると考えられる。また、図 6 において、MN から GS に対して通信開始してから、MN と RS の間にトンネル構築が完了し、NTM 端末と GS の通信が行われるまでにかかる時間は仮想環境にて 268.849ms であった。図 6 に示す NTM Mobile のトンネル構築シーケンスにおいて、NTM Direction Request, DNS Request / Response, NTM Relay Direction / Response の送信には各 10ms 程度、NTM Route Direction, NTM Tunnel Response の送信には各 60ms 程度の一方向の遅延が発生する。よって、通信開始から NTM 端末と GS の通信が行われるまでは 488.849ms かかると推測できる。

以上のことから、通信を開始して、NTM 端末と GS との通信を行うまでの時間のうち、ホップ数調査が占める時間は 12.5%程度であり、実用上の影響はないと考えられる。

6. まとめ

本論文では、NTM 端末同士、及び NTM 端末と GS の通信時に、RS を経由することによる通信経路冗長化の解決手法について提案を行った。NTM 端末同士の通信の場合は、NTM 端末が起動時に、NTM 端末と RS の間のホップ

表 2 仮想マシンの構成

| | DC, MN, RSA, RSB, RSC, DNS, GS | Router |
|--------------|--------------------------------|-----------------------------|
| OS | Ubuntu 10.04 | Ubuntu 10.04 |
| Linux Kernel | 2.6.32-24-generic | 2.6.32-24-generic |
| CPU | Intel Core i7-2600(3.40GHz) | Intel Core i7-2600(3.40GHz) |
| Memory | 各 1GB | 各 512MB |

数の調査を行い、ホップ数が最少となる RS を選択することによって、最短経路での通信を実現できる。また、NTM 端末と GS の通信の場合は、NTM 端末と GS の通信開始時に、RS と GS の間のホップ数を調査し、ホップ数が最少となる RS を選択することによって、NTM 端末が移動した後も最短経路での通信を実現することが可能となる。提案方式では、端末や接続したネットワークの構成に関わらず、通信端末から RS までのホップ数調査を行う手法を確立した。今後は NTM 端末同士で通信を行う場合のホップ数調査を実装し、動作確認を行う予定である。

謝辞 本研究は JSPS 科研費基盤研究 (C)15K00140、及び日本私立学校振興・共済事業団平成 27 年度学術研究振興資金（若手研究者奨励金）の助成を受けたものである。

参考文献

- [1] Perkins, C.: IP Mobility Support for IPv4, Revised, RFC 5944, IETF (2010).
- [2] Perkins, C., Johnson, D. and Arkko, J.: Mobility Support in IPv6, RFC 6275, IETF (2011).
- [3] Soliman, H.: Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555, IETF (2009).
- [4] 鈴木秀和, 上醉尾一真, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃: NTMobile における通信接続性の確立手法と実装, 情報処理学会論文誌, Vol. 54, No. 1, pp. 367-379 (2013).
- [5] 内藤克浩, 上醉尾一真, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊 晃, 森香津夫, 小林英雄: NTMobile における移動透過性の実現と実装, 情報処理学会論文誌, Vol. 54, No. 1, pp. 380-397 (2013).
- [6] 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃: IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価, 情報処理学会論文誌, Vol. 54, No. 10, pp. 2288-2299 (2013).
- [7] 土井敏樹, 鈴木秀和, 内藤克浩, 渡邊 晃: NTMobile におけるアドレス変換型リレーサーバの実装と動作検証, 情報処理学会研究報告, Vol. 2013-MBL-67, No. 11, pp. 1-6 (2013).
- [8] Montenegro, G.: Reverse Tunneling for Mobile IP, revised, RFC 3024, IETF (2001).
- [9] Levkowetz, H. and Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices, RFC 3519, IETF (2003).
- [10] Kulkarni, M., Patel, A. and Leung, K.: Mobile IPv4 Dynamic Home Agent (HA) Assignment, RFC 4433, IETF (2006).
- [11] Abley, J. and Lindqvist, K.: Operation of Anycast Services, RFC 4786, IETF (2006).
- [12] D.Johnson: Mobility Support in IPv6, Technical report (2004).
- [13] J.Arkko: Enhanced Route Optimization for Mobile IPv6, RFC 4866, IETF (2007).
- [14] 納堂博史, 鈴木秀和, 内藤克浩, 渡邊 晃: NTMobile における自律的経路最適化の提案, 情報処理学会論文誌, Vol. 54, No. 1, pp. 394-403 (2013).
- [15] CAIDA: Archipelago Monitor Statistics, <http://www.caida.org/projects/ark/statistics/> (2015 年 1 月 26 日アクセス).