

平成26年度 卒業論文

和文題目

**NTMobile**における  
最適なリレーサーバを選択する手法の提案

英文題目

**Proposal of Selection Method of  
Optimal Relay Server in NTMobile**

情報工学科 渡邊研究室  
(学籍番号: 110430115)

三宅 佑佳

提出日: 平成27年2月12日

名城大学理工学部



## 内容要旨

移動通信端末の普及や無線通信技術の発展により、ネットワーク環境によらず自由に通信できる通信接続性と、通信中にネットワークを切り替えても通信を継続できる移動透過性が求められている。通信接続性と移動透過性を同時に実現する技術として、我々はNTMobile (Network Traversal with Mobility) を提案している。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。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。また、提案方式の性能評価を行い、RS 選択手法によって通信端末間において最適な通信経路を構築できることを確認した。



# 目次

第 1 章	はじめに	1
第 2 章	既存研究	3
2.1	Mobile IPv4	3
2.1.1	Mobile IPv4 の概要	3
2.1.2	Mobile IPv4 の課題	3
2.2	Mobile IPv6	4
2.2.1	Mobile IPv6 の概要	4
2.2.2	Mobile IPv6 の課題	4
2.3	Dual Stack Mobile IPv6	5
2.3.1	DSMIP の概要	5
2.3.2	DSMIP の課題	5
第 3 章	NTMobile	7
3.1	NTMobile の概要	7
3.2	NTMobile の構成	7
3.3	NTMobile の動作	8
3.3.1	NTM 端末のアドレス情報登録処理	8
3.3.2	NTM 端末同士の通信におけるトンネル構築処理	8
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	9
3.4	RS を経由する通信経路の課題	11
第 4 章	提案方式	12
4.1	RS の評価指標	12
4.2	ホップ数調査	13
4.2.1	NTM 端末同士の通信におけるホップ数調査	13
4.2.2	NTM 端末と一般サーバとの通信におけるホップ数調査	13
4.3	RS の選択	14
4.3.1	同一の DC に管理されている NTM 端末同士の通信	14
4.3.2	異なる DC に管理されている NTM 端末同士の通信	15
4.3.3	NTM 端末と一般サーバとの通信	17

<b>第 5 章</b>	<b>実装と評価</b>	<b>20</b>
5.1	実装 . . . . .	20
5.1.1	DC への実装 . . . . .	20
5.1.2	RS への実装 . . . . .	20
5.1.3	NTM 端末への実装 . . . . .	21
5.2	動作検証 . . . . .	22
5.3	性能評価 . . . . .	23
<b>第 6 章</b>	<b>既存研究との比較</b>	<b>25</b>
6.1	IPv4 ネットワークにおける比較 . . . . .	25
6.2	IPv6 ネットワークにおける比較 . . . . .	26
6.3	IPv4/IPv6 混在ネットワークにおける比較 . . . . .	27
<b>第 7 章</b>	<b>まとめ</b>	<b>29</b>
	謝辞	31
	参考文献	33
	研究業績	35
	付録 A メッセージフォーマット	37
	付録 B 各 OS の TTL 初期値	41

# 第1章 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、ネットワークにおける高トラフィックを回避する為、複数の無線回線に分散させたいという要望、更に、移動しながら通信を行いたいという要求が高まっている。

現在、IP ネットワークでは、通信端末に割り当てられた IP アドレスが各端末の通信識別子となっている。通信端末の移動によりネットワークが切り替わると IP アドレスが変化する為、通信を継続することができない。その為、ネットワークを切り替えても通信を継続できる移動透過性の実現が要求されている。また、現在の主流である 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。短期的な解決策として、NAT (Network Address Translation) の導入が挙げられる。NAT 配下の通信端末にプライベートアドレスを割り当て、NAT の機能により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で、通信を行う。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 NAT 越え問題を生じさせ、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、膨大なアドレス空間を持つ IPv6 ネットワークが導入されているが、IPv6 アドレスには IPv4 アドレスとの互換性が無い為、普及が滞っている。その為、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できる通信接続性の実現が期待されている。

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兼ねる通信中継装置 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する。しかし、通信中継装置を経由する通信は経路が冗長になる為、通信経路冗長化の抑制を考慮した中継装置の選択が必要である。

我々は、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 [4-7]。NTMobile は基本的に端末間でエンドツーエンドの直接通信を行うが、通信相手端末が一般サーバの場合等、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。

以後、2章では既存技術について、3章では NTMobile 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．6章では既存研究との比較を行い，最後に7章でまとめる．

## 第2章 既存研究

本章では、通信接続性と移動透過性を同時に実現する既存の技術について述べる。

### 2.1 Mobile IPv4

#### 2.1.1 Mobile IPv4 の概要

MIPv4 は、IPv4 ネットワークを対象とした移動透過技術である。図 2.1 に MIPv4 のネットワーク構成を示す。MIPv4 では、MIPv4 の機能を持つ移動通信端末 MN (Mobile Node) と、ホームネットワークに設置したアドレスの管理と通信の中継を行う装置 HA との間にトンネルを構築し、通信を行う。MIPv4 では、MN が端末識別子として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 は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為、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。

MN は利用する HA を動的に選択可能であることが定義されている [8]。MN が、HA の割り当てを実施する HA (Requested HA) に対して HA 割当要求を送信し、Requested HA が割り当てられた HA (Assigned HA) のアドレスを HA に対して返信する。そして、MN が Assigned HA に対して位置登録を実施することで動的な HA の選択が行われる。

また、MIPv4 は NAT 越えが実現可能である [9]。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。

#### 2.1.2 Mobile IPv4 の課題

MIPv4 では、HA を常に経由した通信になる為、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の増大が懸念される。HA は HoA 宛てのパケットを代理受信する為、HA の

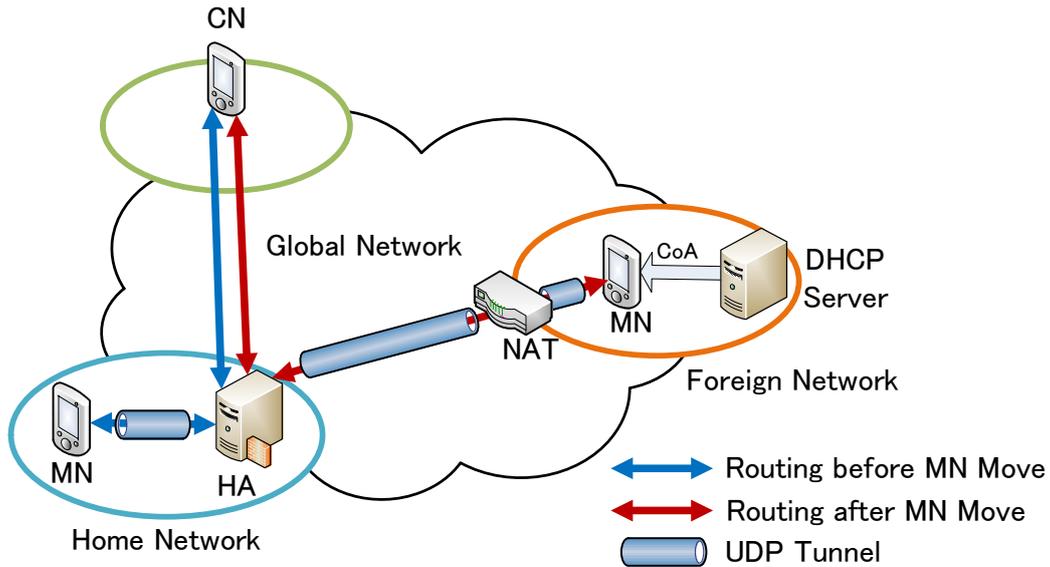


図 2.1 Mobile IPv4 のネットワーク構成

設置箇所はホームネットワーク内に限定されている。その為、グローバルネットワーク上に分散配置が不可能であり、負荷分散が困難である。MIPv4 では動的 HA 選択処理が定義されているが、Requested HA が複数の HA を評価し、Assigned HA を決定する具体的な手法は定められていない。また、HA の設置場所は必ずグローバル上に設置される為、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 アドレス枯渇問題に逆行する致命的な課題である。

## 2.2 Mobile IPv6

### 2.2.1 Mobile IPv6 の概要

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 2.2 に MIPv6 のネットワーク構成を示す。MIPv6 は MIPv4 と同様、ホームネットワークに設置された HA から割り当てられた HoA を用いて通信を行う。MN は移動後、移動先ネットワークから CoA を取得し、HA に登録を行う。パケットは MN と HA との間に構築された IP トンネルを用いて CoA 宛てのカプセル化パケットとして転送される。MN がパケットのデカプセル化を行い、MN のアプリケーションは HoA 宛てのパケットを受け取る。

MIPv6 では、エニーキャスト [10] を用いた Dynamic Home Agent Address Discovery という HA 動的発見手法が定義されている [11]。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為、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]。

### 2.2.2 Mobile IPv6 の課題

MIPv6 は MIPv4 と同様、HA をホームネットワーク内に設置しなければならないという制約がある。また、MIPv6 では HA を選択できる方式が定義されているが (RFC3775)、通信中は HA を

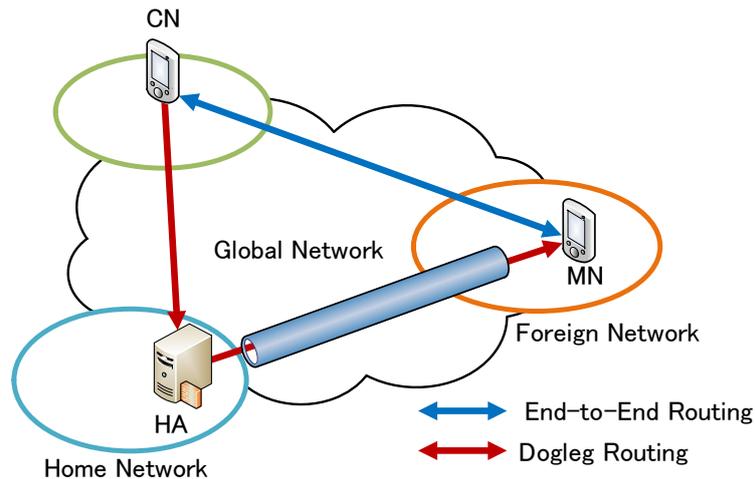


図 2.2 Mobile IPv6 のネットワーク構成

切り替えることができない。従って，端末移動後の通信経路が冗長になる可能性があり，スループットの低下や，ネットワーク負荷の増大が懸念される。

## 2.3 Dual Stack Mobile IPv6

### 2.3.1 DSMIP の概要

図 2.3 に DSMIP のネットワーク構成を示す。DSMIP は，MIPv4 と MIPv6 を組み合わせた移動透過性技術である。DSMIP のホームネットワークは，IPv4/IPv6 混在環境でも動作可能なデュアルスタック構成であり，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。MN は IPv4/IPv6 の HoA を取得可能であり，HA が双方のネットワークの橋渡しを行い，混在環境でも通信を行うことができる。また，NAT 越えは，HA と NAT 配下の MN との間に UDP トンネルを構築することによって実現される。

### 2.3.2 DSMIP の課題

DSMIP は MIPv4 と MIPv6 を単純に結合した技術である為，MIPv4 の課題を引き継いでいる。

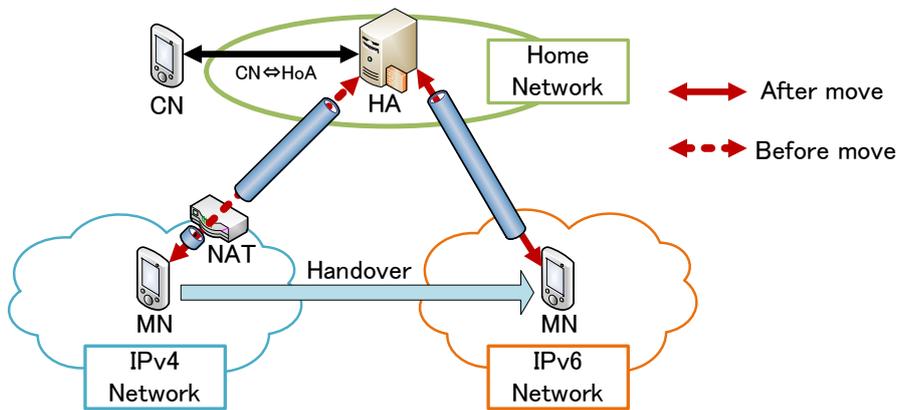


図 2.3 Dual Stack Mobile IPv6 のネットワーク構成

## 第3章 NTMobile

本章では、提案方式を適用する NTMobile の概要と課題について述べる。

### 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。仮想 IP アドレスは、端末の移動により変化せず、実ネットワークに依存しないアドレスである。その為、通信中の端末が移動した場合でも、IP アドレスの変化を通信相手端末 CN や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。

### 3.2 NTMobile の構成

図 3.1 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。

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 端末は仮想 IP アドレスを端末識別子として利用する。また、NTM 端末の実 IP アドレスは位置識別子として利用される。NTM 端末のアプリケーションは割り当てられた仮想 IP アドレスを自身の IP アドレスと認識して通信を行う。

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。直接通信できない通信環境は、NAT 配下に存在する NTM 端末どうしの通信の場合、NTM 端末と NTMobile を実装していない一般サーバ GS (General Server) 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 NTM 端末から通信を開始することができる。また、RS を経由する通信において、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 の場合、経路最適化が適用されない為、RS を経由した通信となる。

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、通信時に利用する RS を必要に応じて自由に選択し、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自由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、DC は RS の負荷情報を収集し、RS の負荷分散を行うことができる [15]。

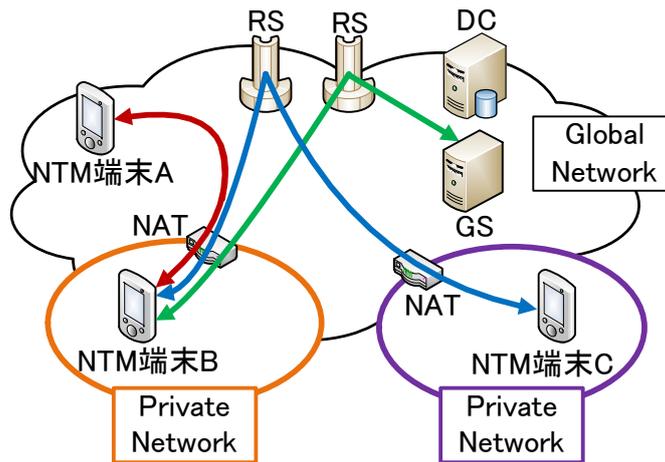


図 3.1 NTMobile のネットワーク構成

### 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 Registration Request を受信すると、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。アドレス情報の登録が完了した後は、NTM 端末と DC との間で定期的なメッセージの交換 (Keep Alive) を行うことにより、DC と NAT 配下に存在する NTM 端末との間の制御メッセージ用の通信経路を確保する。

#### 3.3.2 NTM 端末同士の通信におけるトンネル構築処理

図 3.2 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。なお、MN と CN は異なる NAT 配下に存在するとする。

はじめに、MN は DC<sub>MN</sub> に対して CN の名前解決及びトンネル構築の依頼をする為、NTM Direction Request を送信する。NTM Direction Request には、CN の FQDN (FQDN<sub>CN</sub>) を記載する。DC<sub>MN</sub> 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為、CN を管理している DNS サーバ DNS<sub>CN</sub> に対して、DNS Request for TXT Record を送信し、TXT レコード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為、DC<sub>MN</sub> は DNS<sub>CN</sub> から DNS Request for TXT Record が返ると、DNS<sub>CN</sub> が DC (DC<sub>CN</sub>) であり、CN が NTM 端末であると判明する。DC<sub>MN</sub> は CN が NTM 端末であることがわかると、FQDN<sub>CN</sub> を記載した NTM Information Request を送信し、それを受信した DC<sub>CN</sub> は NTM Information Response に CN の端末情報を載せ、DC<sub>MN</sub> へ応答を返す。

DC<sub>MN</sub>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配下に存在する為、DC<sub>MN</sub>はRSを経由して通信を行うことを決定する。DC<sub>MN</sub>は自身の管理下にあるRS<sub>MN</sub>に対し、MNとの間、及びCNとの間にトンネル構築を行うよう指示する為、NTM Relay Directionを送信する。NTM Relay Directionを受信したRS<sub>MN</sub>は、トンネル構築が可能であること、すなわち自身を中継した通信ができることをDC<sub>MN</sub>に伝える為、NTM Relay ResponseをDC<sub>MN</sub>に送信する。DC<sub>MN</sub>はNTM Route Directionを、MNに対しては直接、CNに対してはDC<sub>CN</sub>の管理下にある為、DC<sub>CN</sub>を経由して送信する。これにより、DC<sub>MN</sub>は、MNとCNに対してRS<sub>MN</sub>との間にトンネル構築を行うよう指示する。指示を受けたMN及びCNは、RS<sub>MN</sub>とトンネル構築する為、NTM Tunnel RequestをRS<sub>MN</sub>に対して送信する。その後、MN-RS<sub>MN</sub>間、及びRS<sub>MN</sub>-CN間にトンネルを構築する。

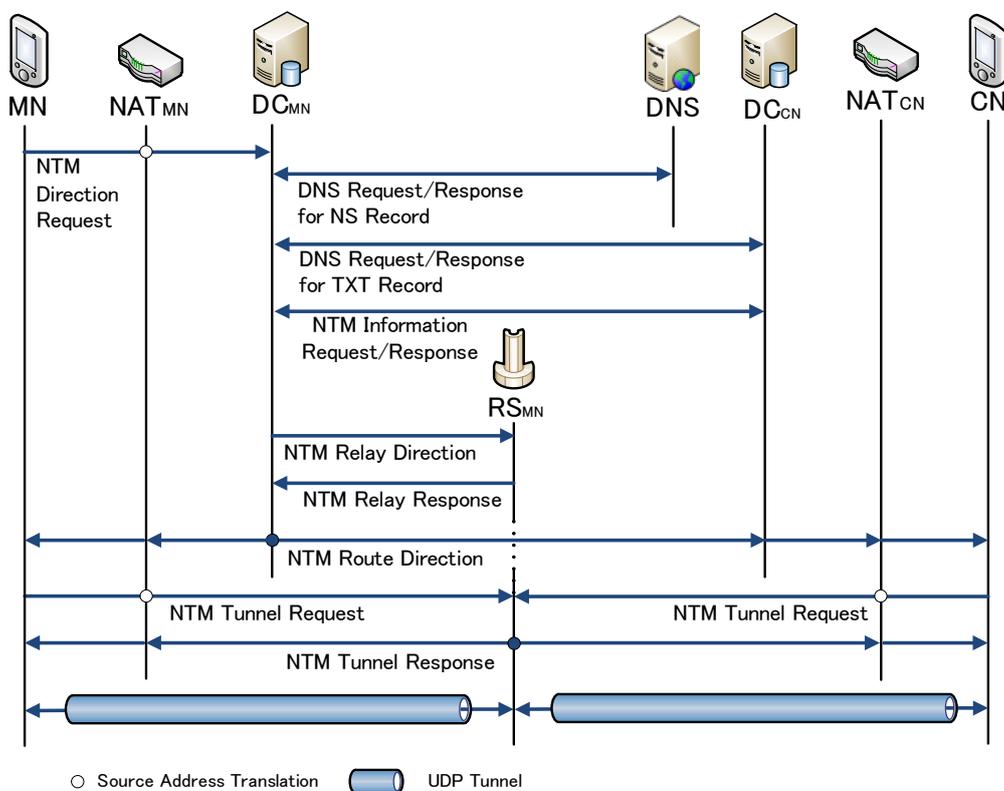


図 3.2 NTM 端末同士のトンネル構築シーケンス

### 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 端末から GS に対して通信を行う際、GS は RS からの通信であると認識する。NTM 端末の移動透過性を確保する為、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。

図 3.3 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 DC<sub>MN</sub> に対して GS の名前解決及びトンネル構築依頼の為、NTM Direction Request を送信する。依頼を受けた DC<sub>MN</sub> は DNS Request / Response for NS Record により、GS の NS レコードを取得

する。次に、DNS<sub>GN</sub> に対して DNS Request for TXT Record を送信し、TXT レコードの問い合わせを行う。DNS<sub>GN</sub> には自身が DC であることを示す TXT レコードが登録されていない為、DNS Response for TXT Record を受信した DC<sub>MN</sub> は通信相手が GS であると判断する。そして、DC<sub>MN</sub> は DNS Request / Response for A / AAAA Record により、GS のアドレス情報を収集する。DC<sub>MN</sub> は GS のアドレス情報を取得すると、その情報を記載した NTM Relay Direction を RSMN に送信し、MN との間でトンネル構築し、通信の中継を行うよう指示する。NTM Relay Direction を受信した RSMN はトンネル構築可能であることを DC<sub>MN</sub> に伝える為、NTM Relay Response を返信する。その後、DC<sub>MN</sub> 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為、NTM Route Direction を送信し、その指示を受け取った MN は RSMN に対して NTM Tunnel Request を送信し、RSMN との間においてトンネル構築を行う。

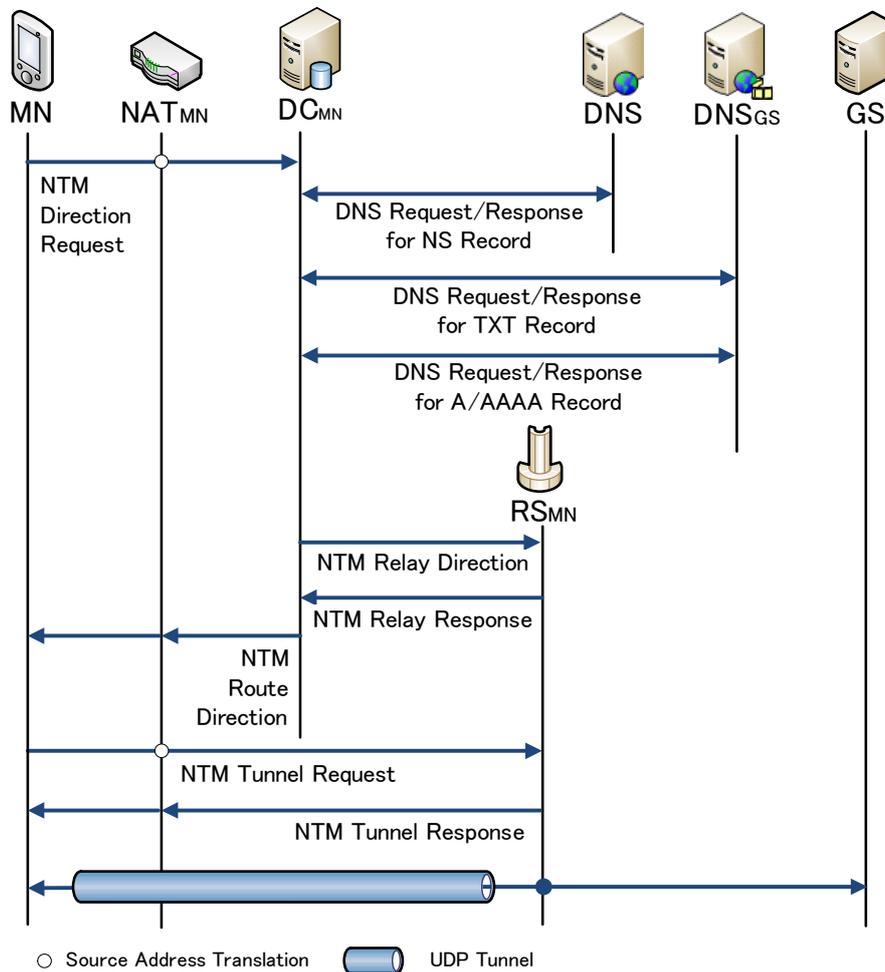


図 3.3 NTM 端末と一般サーバとのトンネル構築シーケンス

### 3.4 RSを経由する通信経路の課題

通信端末間での最適な通信経路は、通信経路において冗長性のない直接通信である。しかし、直接通信ができない場合はRSを経由した通信になり、直接通信を行う場合と比較すると通信経路が冗長になる。

NTM 端末同士の通信の場合、基本的に端末間の直接通信を行うことができるが、NTM 端末が異なる NAT 配下にある場合、IPv4/IPv6 の混在環境下にある場合はRSを経由する通信となる。その為、適切なRSが選択されない場合、通信を行うNTM 端末からネットワーク上の位置が遠いRSを経由した通信になる可能性がある。

NTM 端末と一般サーバとの通信の場合、必ずRSを経由した通信となる。また、一般サーバは通信相手をRSとしてNTM 端末と通信を行う為、通信中にRSを切り替えることができない。その為、通信開始時に適切なRSが選択されない場合、NTM 端末と一般サーバの間の通信経路が冗長になる可能性がある。更に、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。その為、図 3.4 に示すように、通信時において通信経路の冗長性について、且つNTM 端末が移動した後について考慮したRSを選択する手法を確立する必要がある。

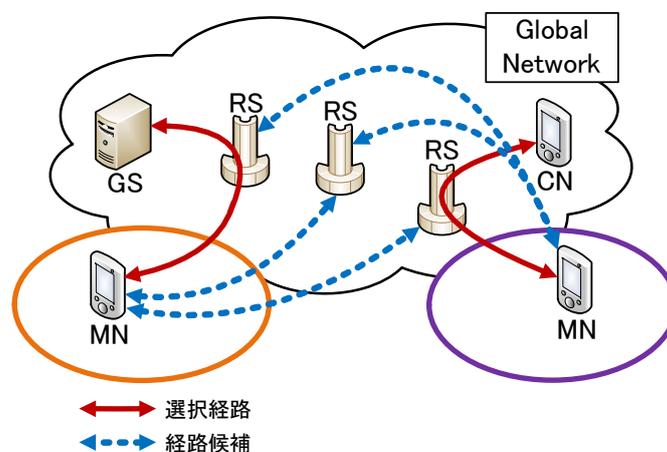


図 3.4 通信経路の冗長性を考慮した通信経路選択

## 第4章 提案方式

本章では、提案する RS 選択手法について述べる。

通信経路の冗長化を抑制する為、RS の負荷分散を行うとともに、通信経路のルータ経由数（ホップ数）を最少とする RS を選択する手法を提案する。NTM 端末同士の通信の場合、NTM 端末が起動した際、NTM 端末から RS までのホップ数調査を行う。そして、通信が開始したときに、その結果を基にして NTM 端末から各 RS までにあるホップ数を算出し、その中から最適な RS を選択する [16]。NTM 端末と一般サーバの通信の場合、NTM 端末から一般サーバへの通信を開始した際、DC は RS から一般サーバまでのホップ数調査を行う。そして、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 を選択し、通信経路を構築する。

### 4.1 RS の評価指標

RS の評価は、NTM 端末から各 RS までの通信経路のホップ数と負荷情報により行う。IPv4 ネットワークにおけるホップ数は TTL (Time to Live)、IPv6 ネットワークにおけるホップ数は Hop limit を用いて調査する。TTL、Hop limit は IP パケットがルータを経由するごとに値が減少する為、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) が挙げられる。NTM 端末が接続するネットワークは無線環境であることが想定されるが、回線の帯域の狭いネットワークにおいては、調査用制御メッセージを多数送受信することは望ましくない。また、帯域の狭いネットワークでは RTT が比較的長く、振れ幅が大きい為、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。これは、ネットワークと端末にかかる負荷が増大し、NTM 端末が移動を行うほど影響が大きくなる為、RTT は評価指標として適さない。一方、ホップ数は通信経路の環境に依存する指標である為、接続したネットワークごとに端末間で制御メッセージを 1 つずつ送信するだけでよい。その為、ホップ数の方が評価指標に適している。

CAIDA (The Cooperative Association for Internet Data Analysis) [18] 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。

## 4.2 ホップ数調査

### 4.2.1 NTM 端末同士の通信におけるホップ数調査

図 4.1 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、もしくは移動後にネットワークに接続した際に、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 端末から自身の管理下にある全ての RS までのホップ数調査を開始する。

DC は NTM 端末に対して、調査対象となる RS の IP アド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為のメッセージである NTM Route Survey を送信する。NTM Route Survey には、NTM 端末を識別する Node ID、NTM 端末の TTL 初期値、または Hop limit の初期値、更に調査指示を出した DC の IP アドレスを記載する。各 RS は NTM Route Survey を受信すると、NTM 端末の TTL 初期値、または Hop limit 初期値と、NTM 端末から RS までのルータを経由したことによる TTL または Hop limit の値の変化を確認し、初期値から変化後の値の差分をとることによってホップ数を算出する。また、TTL 初期値、Hop limit 初期値は、OS (Operating System) によって異なる為、NTM Route Survey に記載された、NTM 端末が生成する TTL 初期値、Hop limit 初期値を利用する。RS はホップ数を算出すると、NTM Survey Report に、NTM 端末の Node ID、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 端末から管理下の RS までのホップ数を Hop Table に記録する。

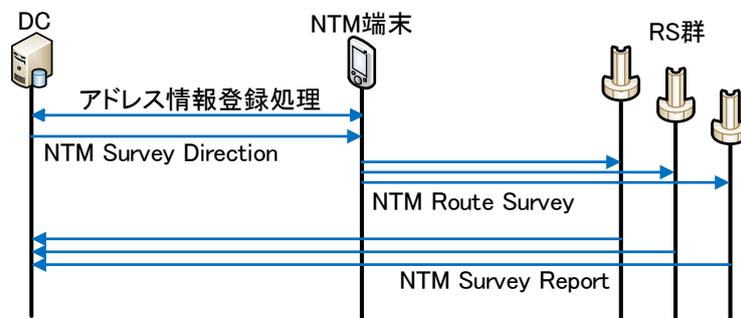


図 4.1 NTM 端末同士の通信におけるホップ数調査シーケンス

### 4.2.2 NTM 端末と一般サーバとの通信におけるホップ数調査

図 4.2 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 は一般サーバの名前解決を行う為、DNS との間で名前解決処理を行う。名前解決処理が行われ、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、もしくは Hop limit の値と GS の TTL、もしくは Hop limit の初期値との差分をとり、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為、算出したホップ数を記載した 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。

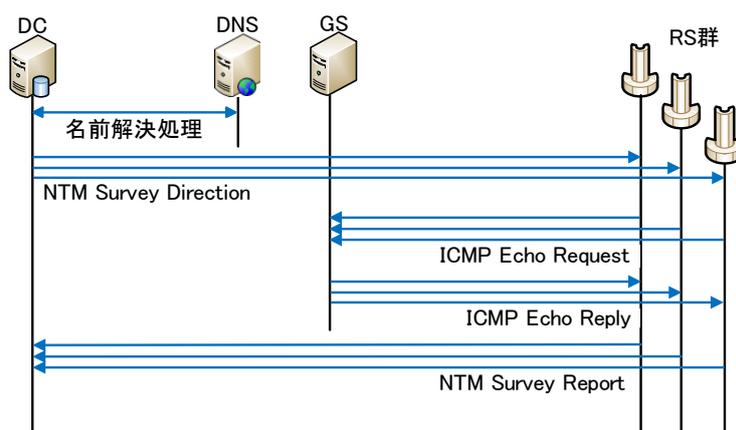


図 4.2 NTM 端末と一般サーバとの通信時におけるホップ数調査シーケンス

### 4.3 RS の選択

RS を必要とする通信は、同一の DC に管理される NTM 端末同士の通信、異なる DC に管理される NTM 端末同士の通信、NTM 端末と一般サーバとの通信に分けられる。これらの通信について、それぞれ通信経路冗長化を抑制する最適な RS の選択を行う。

図 4.3 に同一の DC に管理される NTM 端末同士の通信において想定する NTMobile のシステム構成、図 4.4 に異なる DC に管理される NTM 端末同士の通信において想定する NTMobile のシステム構成を示す。DCMN-CN では、RSx~RSz、DCMN では RSA~RSC、DCCN では RSL~RSN を管理しているとする。また、DC は管理下の RS における負荷分散を考慮する為、RS の負荷情報の収集を行っていることを前提とする。

#### 4.3.1 同一の DC に管理されている NTM 端末同士の通信

図 4.5 に MN と CN が同一の DC に管理されている場合の 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 RS に対してホップ数調査を行い、DCMN-CN の Hop Table に図 4.5 のような調査結果が登録されているとする。また、DCMN-CN は負荷に問題の無い RSx~RSz を選択対象としたとする。

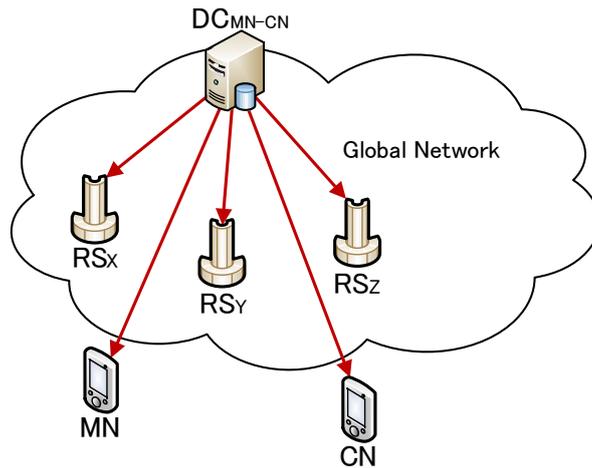


図 4.3 同一の DC に管理される NTM 端末同士の通信におけるシステム構成

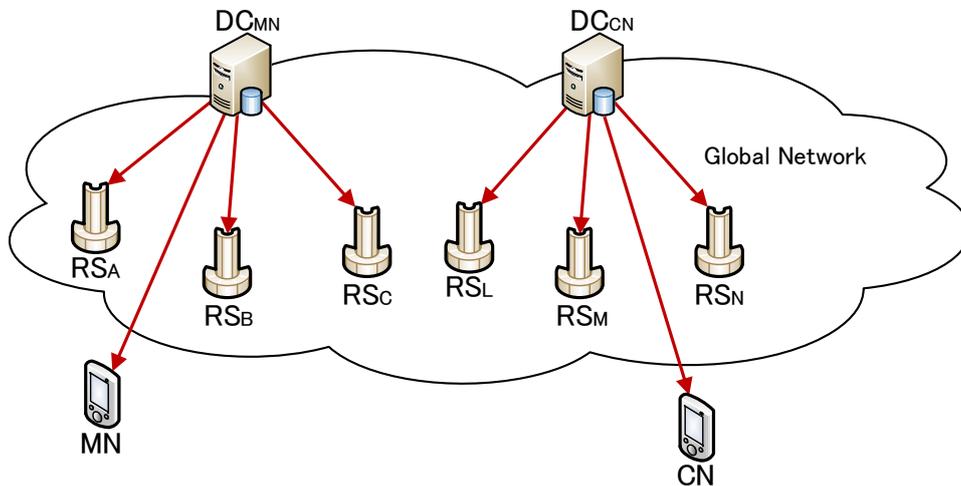


図 4.4 異なる DC に管理される NTM 端末同士の通信におけるシステム構成

DCMN-CN は MN より経路指示依頼のメッセージである 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 RSx までのホップ数と CN から RSx までのホップ数、MN から RSy までのホップ数と CN から RSy までのホップ数、MN から RSz までのホップ数と CN から RSz までのホップ数をそれぞれ合算し、MN から各 RS を経由して CN に到達するまでの総経路ホップ数を算出する。DCMN-CN は算出した各総経路ホップ数の中から、ホップ数が 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。

#### 4.3.2 異なる DC に管理されている NTM 端末同士の通信

MN と CN がそれぞれ DCMN, DC CN に管理された RS の選択手法について、DCMN 配下の RS を選択する場合と、DC CN 配下の RS を選択する場合に分けて説明する。以下の説明では、MN は

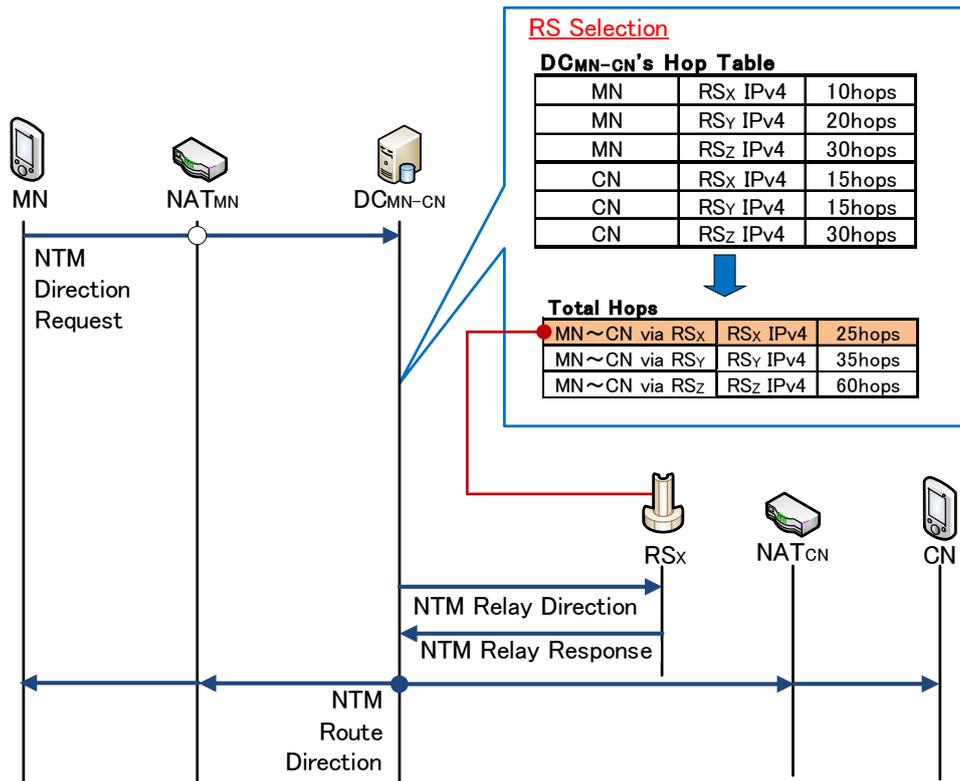


図 4.5 DCMN-CN に管理されている NTM 端末同士の通信時における RS 選択

DCMN 管理下の RS に対して、CN は DCCN 管理下の RS に対してそれぞれホップ数調査を行ったものとし、DCMN と DCCN の Hop Table には図 4.6、図 4.7 のような調査結果が登録されているとする。また、DCMN は管理下の RSA~RSC を、DCCN は管理下の RSL~RSN を選択可能であると判断したとする。

- DCMN 配下の RS を選択する場合

図 4.6 に異なる DC に管理される NTM 端末同士の通信時において、DCMN 管理下の RS を選択する場合のシーケンスを示す。DCMN は MN から CN までの経路指示依頼を受け取ると、DCCN に対して CN の端末情報を取得する為、NTM Information Request を送信する。DCCN は NTM Information Request を受信すると、自身の Hop Table を検索し、選択可能とされた RSL~RSN と CN との間のホップ数を取得する。Hop Table を検索する際のキーは、CN の Node ID、RSL~RSN の IP アドレスを用いる。そして、オプションヘッダである Route Information に RSL~RSN と CN の間のホップ数を記載し、そのオプションヘッダを NTM Information Response に付加して DCMN に送信することで、DCCN が得たホップ数の情報を DCMN に報告する。

DCMN は RSA~RSC と MN との間のホップ数で最少のものと、RSL~RSN と CN との間のホップ数で最少のものを比較し、NTM 端末との間のホップ数が少ない方の RS を選択する。DCMN 管理下の RSA が選択された場合、トンネル構築までの経路指示手順を実施する。

- DCCN 管理下の RS を選択する場合

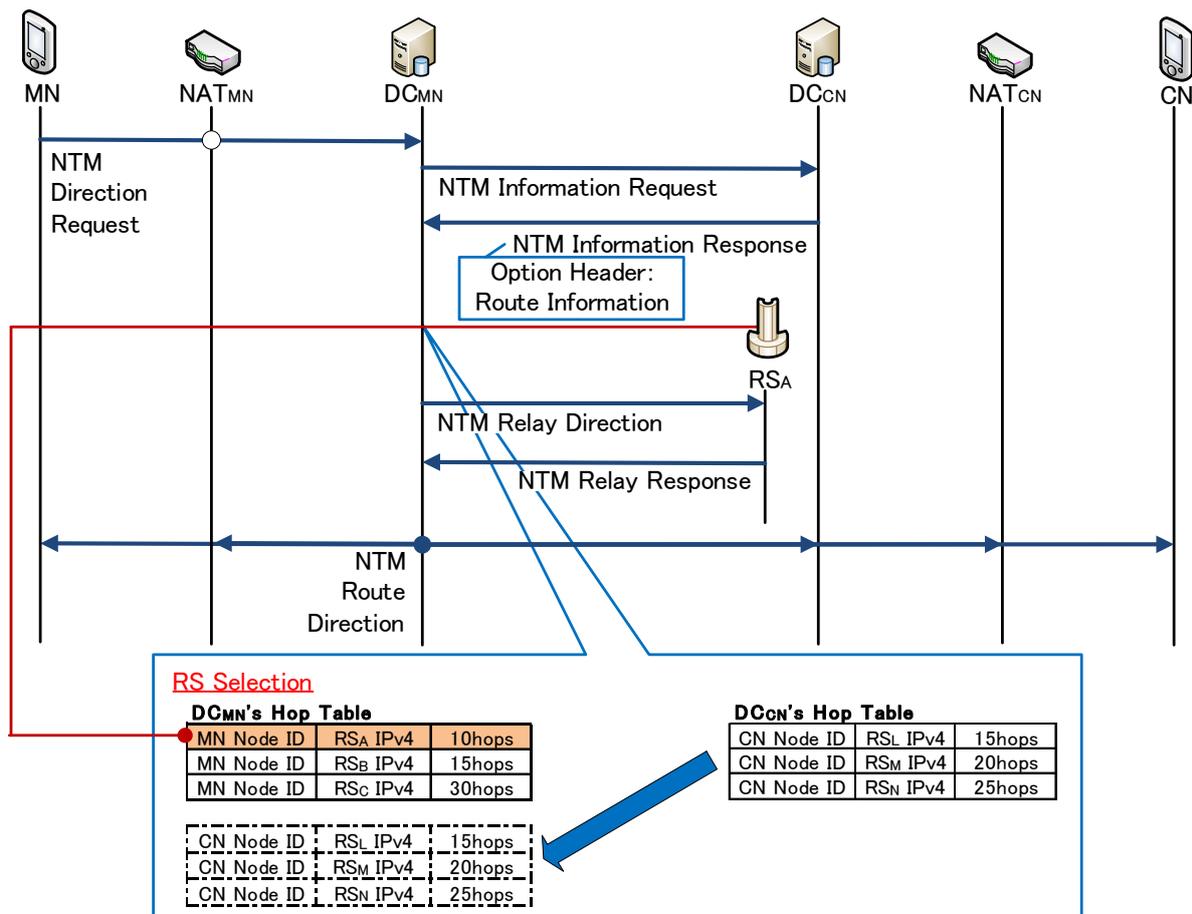


図 4.6 異なる DC に管理される NTM 端末同士の通信時における DCMN 管理下の RS 選択

図 4.7 に異なる DC に管理される NTM 端末同士の通信時において、DCCN 管理下の RS を選択する場合のシーケンスを示す。DCMN 管理下の RS を選択する場合と同様にして、NTM Information Response まで実施し、DCCN が RS<sub>L</sub> を選択したとする。DCCN はトンネル構築の委託を受け、従来通りトンネル構築までの経路指示を実施する。

### 4.3.3 NTM 端末と一般サーバとの通信

図 4.8 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、DNS の仕組みにより DCMN が GS の A/AAAA Record を取得すると、ホップ数調査が開始されるが、図 4.8 ではホップ数調査のシーケンスは省略する。GS は DCMN 管理下にある RS との間のホップ数調査を行い、RS<sub>A</sub>~RS<sub>C</sub> が選択可能と判断されたものとする。

DCMN は DNS との間で GS の名前解決処理を行うと、GS から RS までのホップ数調査を実施する。ホップ数調査後、DCMN は Hop Table の中から最適な RS を選択する。DCMN は Hop Table において、GS の FQDN、RS の IP アドレスをキーとして GS から各 RS までのホップ数の情報を検索し、その中から最もホップ数の少ない RS<sub>A</sub> を選択し、トンネル構築までの経路指示手順を実施する。

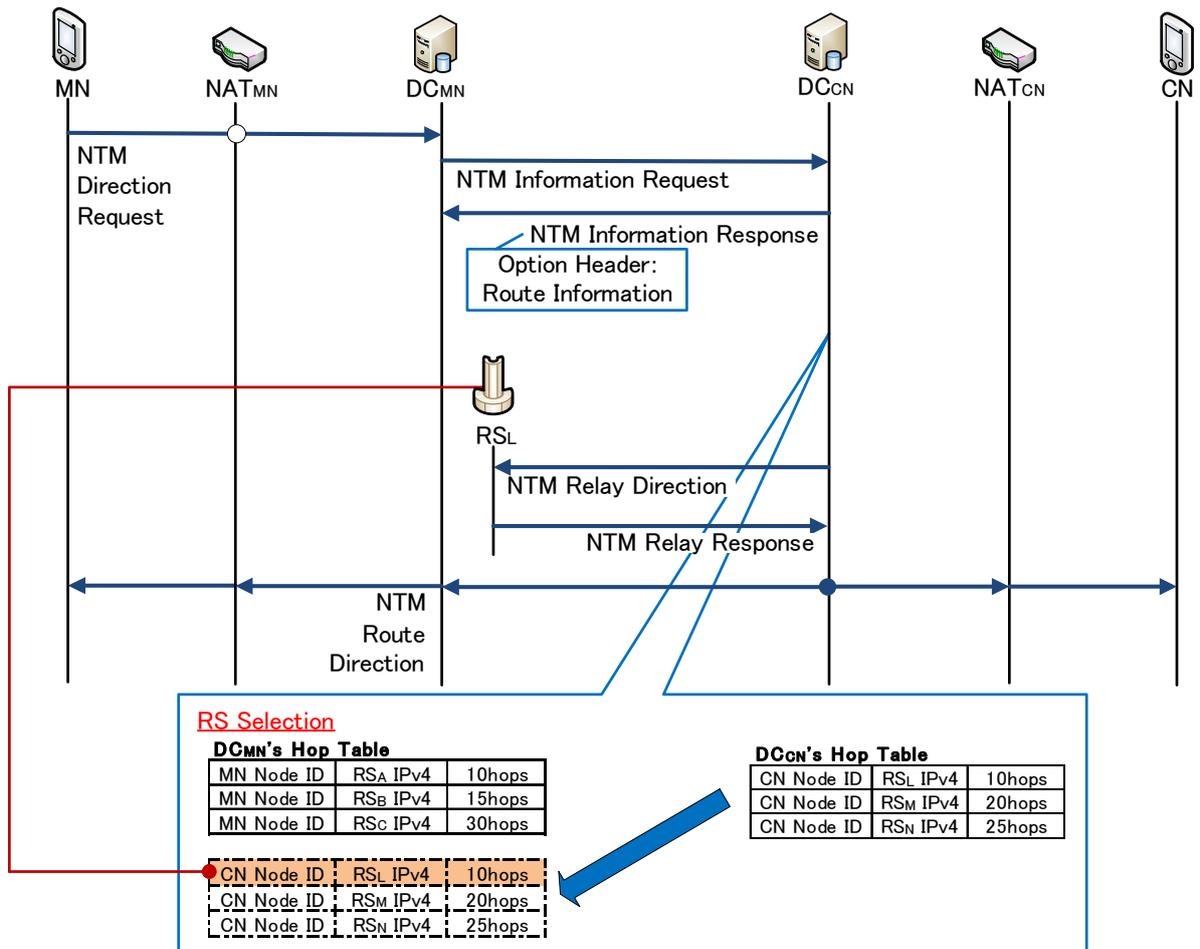


図 4.7 異なる DC に管理される NTM 端末同士の通信時における DCCN 管理下の RS 選択

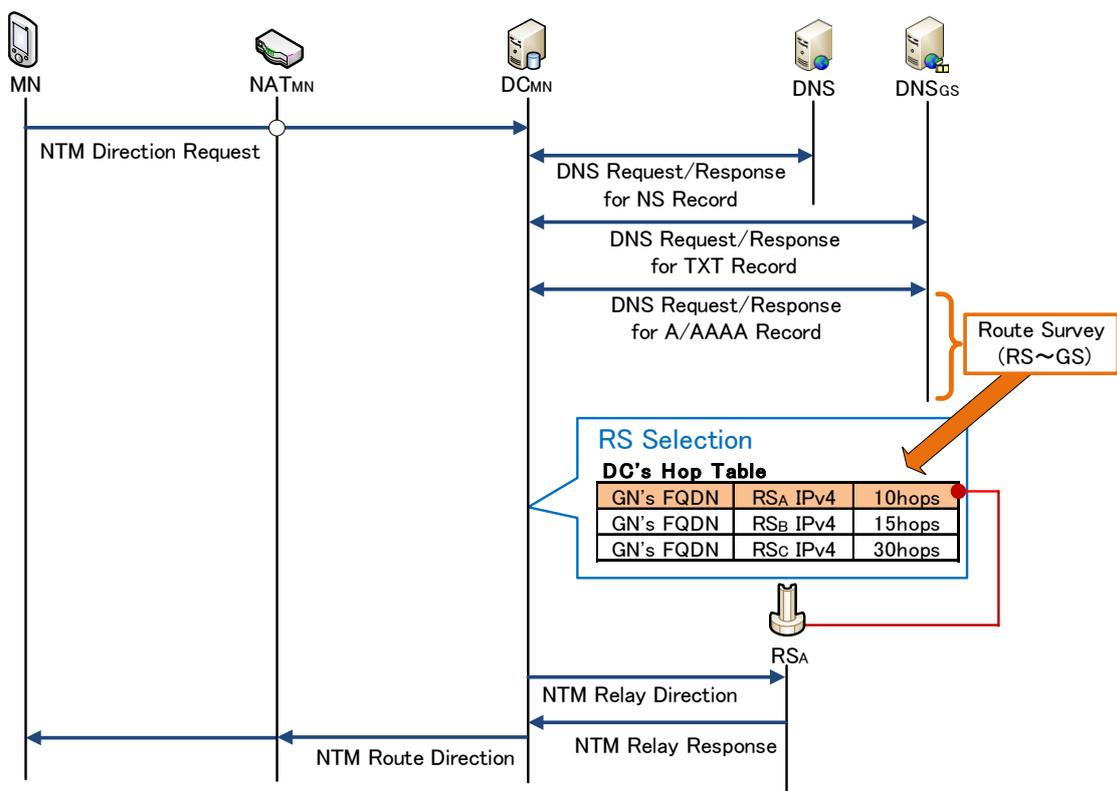


図 4.8 NTM 端末と一般サーバとの通信時における RS 選択

## 第5章 実装と評価

本章では、提案方式の実装とその動作検証、及び性能評価について述べる。

NTMobile 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。提案方式のプロトタイプを DC, RS, NTM 端末へそれぞれ実装を行った。また、NTM 端末と RS の間のホップ数調査の場合と、GS と RS の間のホップ数調査の場合と分けて実装を行った。

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遅延を測定した。

### 5.1 実装

提案方式のプロトタイプを実装した。また、プロトタイプモジュールは IPv4 ネットワークにのみ対応している。

#### 5.1.1 DC への実装

図 5.1 に DC のモジュール構成を示す。DC はユーザ空間の NTM デーモンと、BIND を利用した DNS 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。

RS 選択処理について、NTM 端末同士の通信の場合も NTM 端末と GS の通信の場合も、本来は Hop Table を基にホップ数が最少となる RS を選択する予定である。しかし、本論文では Route Survey の中に最適な RS を選択するようプロトタイプの実装を行った。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 5.2 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。

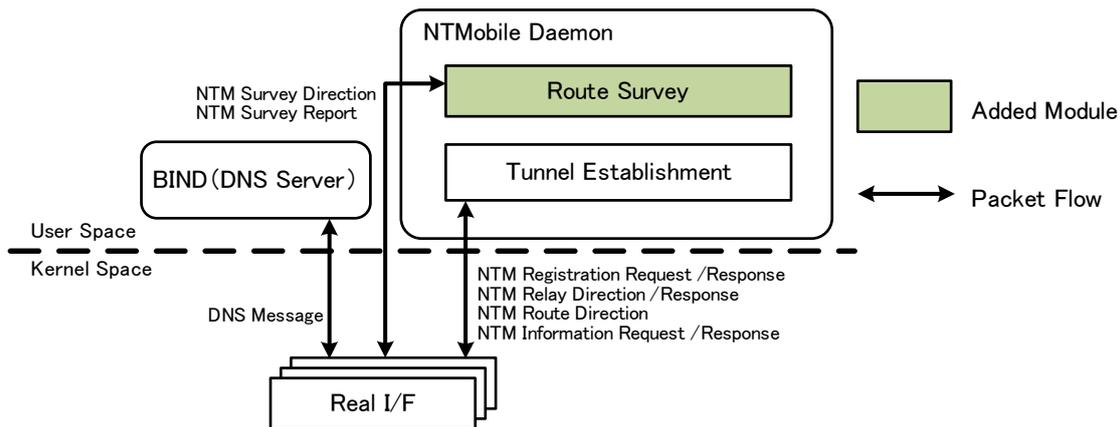


図 5.1 DC のモジュール構成

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。TTL はホップ数を算出する為に必要となるからである。その為、RS ではデバイスレベルの packets 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 packets を受信可能とした。PF\_PACKET を利用することにより、ユーザ空間に限定して実装を行うことができ、実装が容易になる。

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

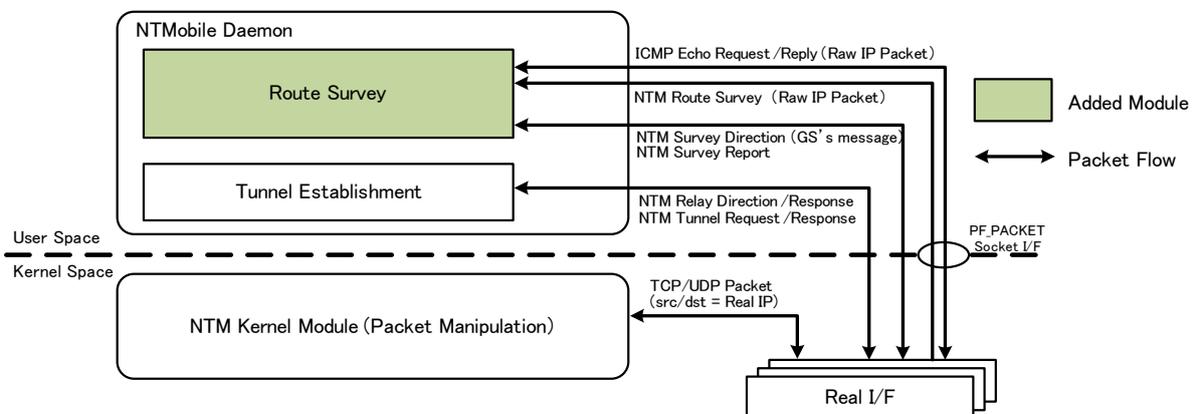


図 5.2 RS のモジュール構成

### 5.1.3 NTM 端末への実装

図 5.3 に NTM 端末のモジュール構成を示す。NTM 端末はユーザ空間の 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り構成される。NTM デーモンは DC に対する NTM 端末情報の登録と仮想 IP アドレスの取得、及びトンネル構築を行う。NTM カーネルモジュールは packets のカプセル化/デカプセル化、及び暗号化処理を行う。NTM デーモンに、NTM 端末と RS の間でホップ数調査を行うモジュール、Route Survey のプロトタイプの実装を行った。

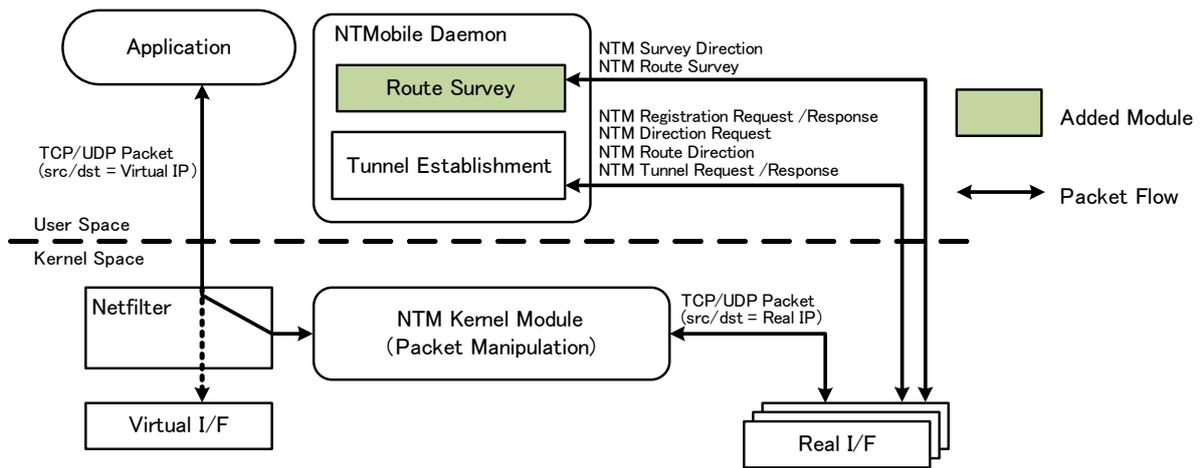


図 5.3 NTM 端末のモジュール構成

## 5.2 動作検証

表 5.1 にホスト PC の構成, 表 5.2 に仮想マシンの構成, 図 5.4 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<sup>\*1</sup> 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 プライベートネットワークに接続し, RS<sub>B</sub> 及び RS<sub>C</sub> とプライベートネットワークの間に 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.

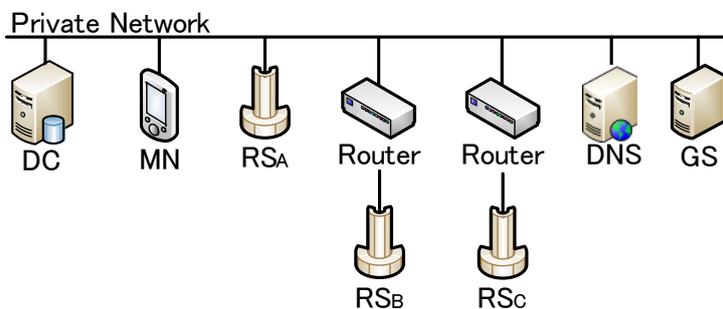


図 5.4 動作検証におけるネットワーク構成

表 5.1 ホスト PC の構成

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

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

表 5.2 仮想マシンの構成

	DC, MN, RSA, RS <sub>B</sub> , RS <sub>C</sub> , 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

### 5.3 性能評価

GS と RS の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 5.5 に、図 5.4 の環境において実行したホップ数調査の時間を示す。ホップ数調査にかかった時間は DC にて Wireshark<sup>\*2</sup> により取得し、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。

図 5.5 は、DC が DNS から DNS Response for A/AAAA Record を受信してから DC が各 RS から NTM Survey Report を受信するまでの時間の内訳を示す。DC が、DNS から DNS Response for A/AAAA Record を受信してから、RS<sub>A</sub> 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、続いて送信された RS<sub>B</sub> に対する NTM Survey Direction は 0.915ms、RS<sub>C</sub> 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<sub>A</sub>, RS<sub>B</sub>, RS<sub>C</sub> 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<sub>A</sub> は、RS<sub>B</sub>, RS<sub>C</sub> と比べてメッセージの送受信に時間がかかっている。これはルータを経由している為である。その為、パケットのルータ経由数が多いほど、遅延の影響が大きくなると考えられる。また、RS<sub>A</sub>, RS<sub>B</sub>, RS<sub>C</sub> 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。この遅延には、各 RS において GS までの間のホップ数を算出する処理時間が含まれている。

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。また、3 章の図 3.3 において、MN から NTM Direction Request の送信により通信開始されてから、MN から RS の間にトンネル構築が完了して NTM 端末と GS の通信が行われるまでにかかる時間は 268.849ms であった。

実環境での調査時間の検討の為、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、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 であるとする。提案する 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 Request, ICMP 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 5.5 の場合における GS から RS までのホップ数調査は、61.12ms で正常に完了する

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

と考えられる。また、図 3.3 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、通信を開始してホップ数調査を行い、NTM 端末と GS との通信を行うまでにおけるホップ数調査時間は全体の 12.5% 程度であり、実用上問題ないと考えられる。

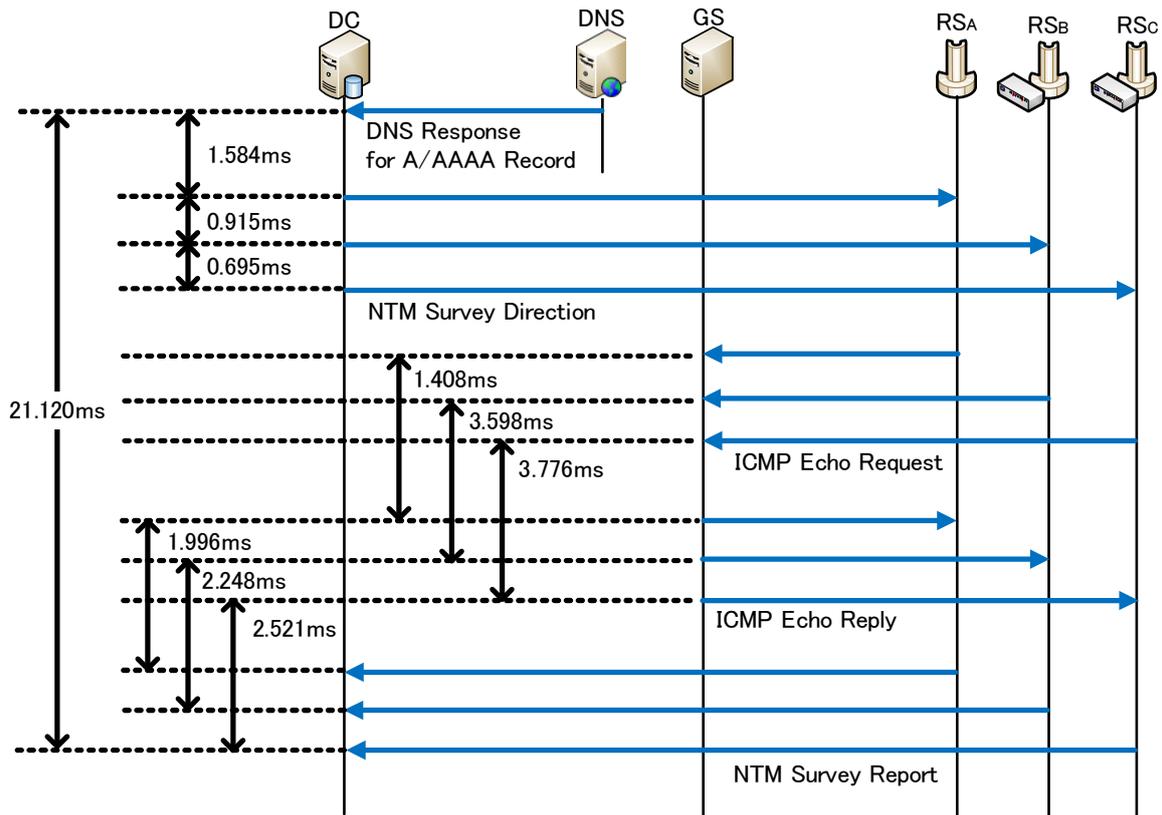


図 5.5 ホップ数調査時間

## 第6章 既存研究との比較

### 6.1 IPv4 ネットワークにおける比較

表 6.1 に IPv4 ネットワークにおける MIPv4, NTMobile の比較を示す。

- 中継装置の分散配置  
MIPv4 では、通信の中継を行う HA を設置する場所がホームネットワーク内に限定される為、分散配置は困難である。一方、NTMobile では中継装置である RS をグローバルネットワーク上に自由に分散配置することができる。
- 端末起動時の中継装置選択  
MIPv4 では、端末起動時に負荷分散、及び通信を行う端末の位置を考慮し、HA を動的に割り当てる。NTMobile では、DC が RS の負荷情報を収集することで、RS の負荷分散が可能である。また、NTMobile は、提案方式によって通信時に最適な RS を選択することができる。
- 端末移動後の中継装置再選択  
MIPv4 では、端末起動時に通信で利用する HA を選択することができるが、端末は HoA に依存した通信を行い、HA が通信端末のアドレス管理を行っている為、端末が移動しても、1つの HA を利用し続ける必要がある。その為、端末移動後に HA を選び直すことができない。NTMobile では、NTM 端末同士の通信時においては通信端末が移動した後も RS を再選択することができる。しかし、NTM 端末と一般サーバの通信の場合は、一般サーバが RS を通信相手と認識して通信を行う為、通信中に RS を切り替えることができない。
- 通信ごとの中継装置選択  
MIPv4 では、上記と同様の理由により、端末起動時以外に HA を選択することができない。NTMobile では、端末の位置に関わらず、通信ごとに RS を選択することが可能である。
- 経路最適化  
MIPv4 においては、通信端末間の経路最適化は定義されていない。しかし、NTMobile では、RS 経由の通信経路構築後、端末間で自律的経路最適化を行うことにより、端末間の直接通信を実現可能である。ただし、両 NAT が Symmetric NAT の場合には経路最適化が行えない為、RS を経由した通信となる。

以上より、MIPv4 では通信経路の冗長化は避けられないが、NTMobile では通信経路冗長化の抑制が可能であると言える。

表 6.1 IPv4 ネットワークにおける提案方式と既存研究との比較

比較項目	MIPv4	NTMobile
中継装置の分散配置	×	○
端末起動時の中継装置選択	○	○
端末移動後の中継装置再選択	×	△
通信ごとの中継装置選択	×	○
経路最適化	×	○
通信経路冗長化の抑制	×	○

## 6.2 IPv6 ネットワークにおける比較

表 6.2 に IPv6 ネットワークにおける MIPv6, NTMobile の比較を示す。NTMobile においては、IPv4 ネットワークの場合と同様の動作を行う。

- 中継装置の分散配置

MIPv6 では、エニーキャストによる HA 動的発見方法が定義されており、経路冗長化の抑制や負荷分散が可能である。しかし、ホームネットワーク内を対象にエニーキャストを用いると、ホームネットワーク内の HA のみが対象となり、負荷分散の効果が小さい。一方、NTMobile では、自由に分散配置することができる。

- 端末起動時の中継装置選択

MIPv6 では、エニーキャストによる HA 動的発見方法により、HA の負荷分散、及び端末から近くにある HA に接続することが可能である。NTMobile では、RS の負荷分散、及び選択が可能である。

- 端末移動後の中継装置再選択

MIPv6 では、経路最適化が適応されず、HA 経由の通信が避けられない場合、MIPv4 と同様に端末が HoA に依存した通信を行う。HA が通信端末のアドレス管理を行っている為、1 つの HA を利用し続ける必要がある。一方、NTMobile では、NTM 端末同士の通信時には RS の再選択が可能であるが、NTM 端末と一般サーバの通信の場合は、RS の再選択を行うことができない。

- 通信ごとの中継装置選択

MIPv6 では、HA の選択手法が定義されていない為、HA を選択することができない。NTMobile では、端末の位置に関わらず、通信ごとに RS を選択することが可能である。

- 経路最適化

MIPv6 では、通信端末間の経路最適化が定義されている。NTMobile では、経路最適化により端末間の直接通信が可能である。

MIPv6 では、エニーキャストや経路最適化により通信経路の冗長化を抑えることが可能である。HA の選択手法は定義されているが、端末移動後や通信ごとに HA を切り替えることができない

為、端末が移動することで経路がより冗長になる。一方、NTMobile では通信経路冗長化の抑制が可能である。

表 6.2 IPv6 ネットワークにおける提案方式と既存研究との比較

比較項目	MIPv6	NTMobile
中継装置の分散配置	△	○
端末起動時の中継装置選択	○	○
端末移動後の中継装置再選択	×	△
通信ごとの中継装置選択	×	○
経路最適化	○	○
通信経路冗長化の抑制	△	○

### 6.3 IPv4/IPv6 混在ネットワークにおける比較

表 6.3 に IPv4/IPv6 混在環境ネットワークにおける DSMIP, NTMobile の比較を示す。

- 中継装置の分散配置
 

DSMIP では、HA をホームネットワーク内に設置する必要がある、分散配置を行うことは難しい。NTMobile では、RS をデュアルスタック構成とした上で、分散配置を行うことができる。
- 端末起動時の中継装置選択
 

DSMIP において、MIPv4 側及び MIPv6 側で HA の動的選択が可能である。NTMobile では、RS の負荷分散、及び選択が可能である。
- 端末移動後の中継装置再選択
 

DSMIP では、MIPv4, MIPv6 と同様、1つの HA を利用し続ける必要がある。一方、NTMobile では、NTM 端末同士の通信時においては RS の再選択が可能であるが、NTM 端末と一般サーバの通信の場合は、RS の再選択を行うことができない。
- 通信ごとの中継装置選択
 

DSMIP では、1つの HA を利用し続ける必要がある為、HA を通信ごとに選択することができない。NTMobile では、端末の位置に関わらず、通信ごとに RS を選択することが可能である。
- 経路最適化
 

DSMIP では、端末が移動した先のネットワークにおいて IPv6 アドレスを取得し、通信相手と IPv6 において通信を行っている場合のみ、経路最適化が適用可能である。一方、NTMobile では、経路最適化により端末間の直接通信が可能である。

以上のことから，DSMIPでは通信経路の冗長は避けられず，NTMobileでは通信経路の冗長を抑えることが可能であると言える。

表 6.3 IPv4 ネットワークにおける提案方式と既存研究との比較

比較項目	DSMIP	NTMobile
中継装置の分散配置	×	○
端末起動時の中継装置選択	○	○
端末移動後の中継装置再選択	×	△
通信ごとの中継装置選択	×	○
経路最適化	△	○
通信経路冗長化の抑制	×	○

## 第7章 まとめ

本論文では，NTM 端末同士，及び NTM 端末と一般サーバの通信時に，中継装置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．提案方式では，ネットワーク負荷を最小限に抑えた方法で，端末のカーネル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．

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 端末と一般サーバの通信の場合は，NTM 端末と一般サーバの通信開始時に，RS と一般サーバ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 端末が移動した後でも最短経路での通信を実現することが可能となる．

提案方式のプロトタイプを DC 及び RS への実装を行った．動作検証を行った結果，IPv4 プライベートネットワークに存在する NTM 端末から RS まで，及び一般サーバから RS までのホップ数調査が正常に完了し，NTM 端末と一般サーバの通信が最適な RS を経由して行われることを確認した．



## 謝辞

本研究を進めるにあたり，終始丁寧かつ熱心なご指導を賜りました，指導教官である名城大学理工学部情報工学科 渡邊晃教授に心から感謝致します。

本研究を進めるにあたり，様々なご指導を頂きました，名城大学理工学部情報工学科 鈴木秀和助教に深く感謝致します。

本研究を進めるにあたり，ご意見並びにご助言を賜りました，愛知工業大学情報科学部情報科学科 内藤克浩助教に深謝致します。

本研究を進めるにあたり，常日頃からご教示下さり，数々のご助言を賜りました，廣瀬達也氏，加古将規氏に心から感謝致します。

最後に，本研究を進めるにあたり，多くの討論の場において有益なご意見を賜りました，渡邊研究室及び鈴木研究室の先輩方，そして同期の皆様に感謝致します。



## 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, 上酔尾一真, 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, 上酔尾一真, 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] 上酔尾一真, 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] Kulkarni, M., Patel, A. and Leung, K.: Mobile IPv4 Dynamic Home Agent (HA) Assignment, RFC 4433, IETF (2006).
- [9] Levkowitz, H. and Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices, RFC 3519, IETF (2003).
- [10] Abley, J. and Lindqvist, K.: Operation of Anycast Services, RFC 4786, IETF (2006).
- [11] D.Johnson: Mobility Support in IPv6, Technical report (2004).
- [12] K.Leung: Proxy Mobile IPv6, RFC 5213, IETF (2008).
- [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] 井貝友哉, 土井敏樹, 上酔尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃: NTMobileにおけるRS-Nの二重化と状態管理手法の提案, 第76回情報処理学会全国大会講演論文集, Vol. 2014, No. 1, pp. 257–258 (2014).
- [16] 若杉 純, 土井敏樹, 鈴木秀和, 内藤克浩, 渡邊 晃: NTMobileにおける通信経路冗長化を抑制するリレーサーバ選択手法の提案, 第76回情報処理学会全国大会講演論文集, Vol. 2014, No. 1, pp. 259–260 (2014).
- [17] CAIDA: Med RTT per Hop,  
[http://www.caida.org/projects/ark/statistics/nrt2-jp/med\\_rtt\\_per\\_hop.csv](http://www.caida.org/projects/ark/statistics/nrt2-jp/med_rtt_per_hop.csv) (2015年1月26日アクセス).
- [18] CAIDA: Archipelago Monitor Statistics, <http://www.caida.org/projects/ark/statistics/> (2015年1月26日アクセス).
- [19] Binbert: Default Time To Live (TTL) values (2015年2月2日アクセス).



# 研究業績

## 研究会・大会等

- (1) 三宅佑佳, 廣瀬達也, 鈴木秀和, 内藤克浩, 渡邊 晃 : NTMobile における最適なりレーサーバ選択手法の提案, 平成 26 年度電気関係学会東海支部連合大会論文集, No.P4-1(2014).
- (2) 三宅佑佳, 鈴木秀和, 内藤克浩, 渡邊 晃 : NTMobile 端末と一般サーバとの通信時に通信経路冗長化を抑制するりレーサーバ選択手法の提案, 2015 年電子情報通信学会総合大会講演論文集(2015).



# 付録A メッセージフォーマット

図 A.1 に NTMobile のメッセージに付加される NTM ヘッダとヘッダオプションのフォーマットを示す。

また、図 A.2～図 A.4 に 4 章において提案したメッセージフォーマットを示す。

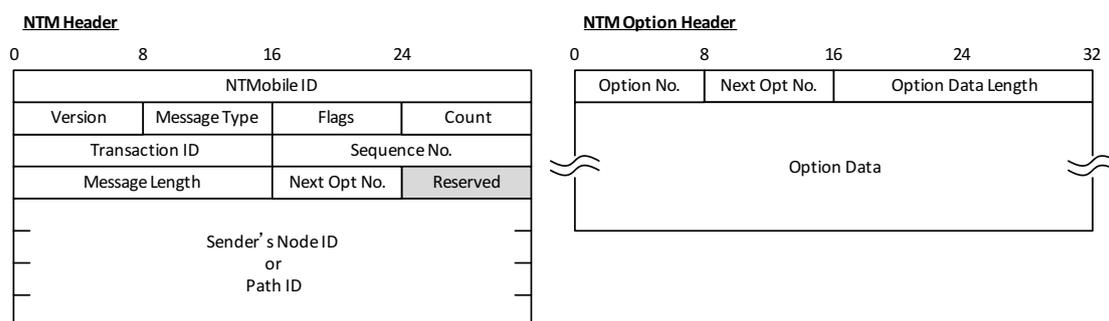


図 A.1 NTM ヘッダと NTM ヘッダオプション

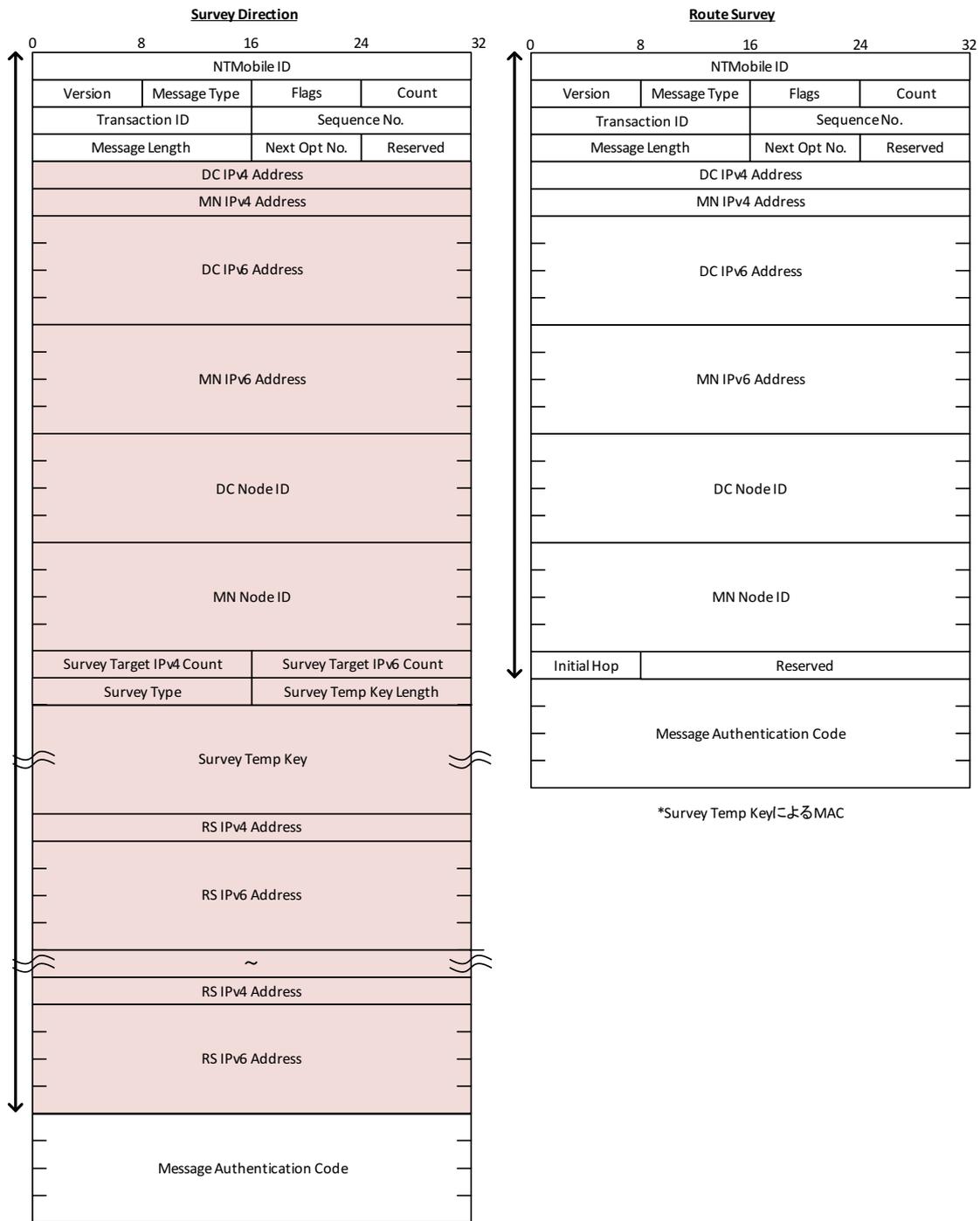


図 A.2 NTM 端末と RS のホップ数調査における提案のメッセージフォーマット 1

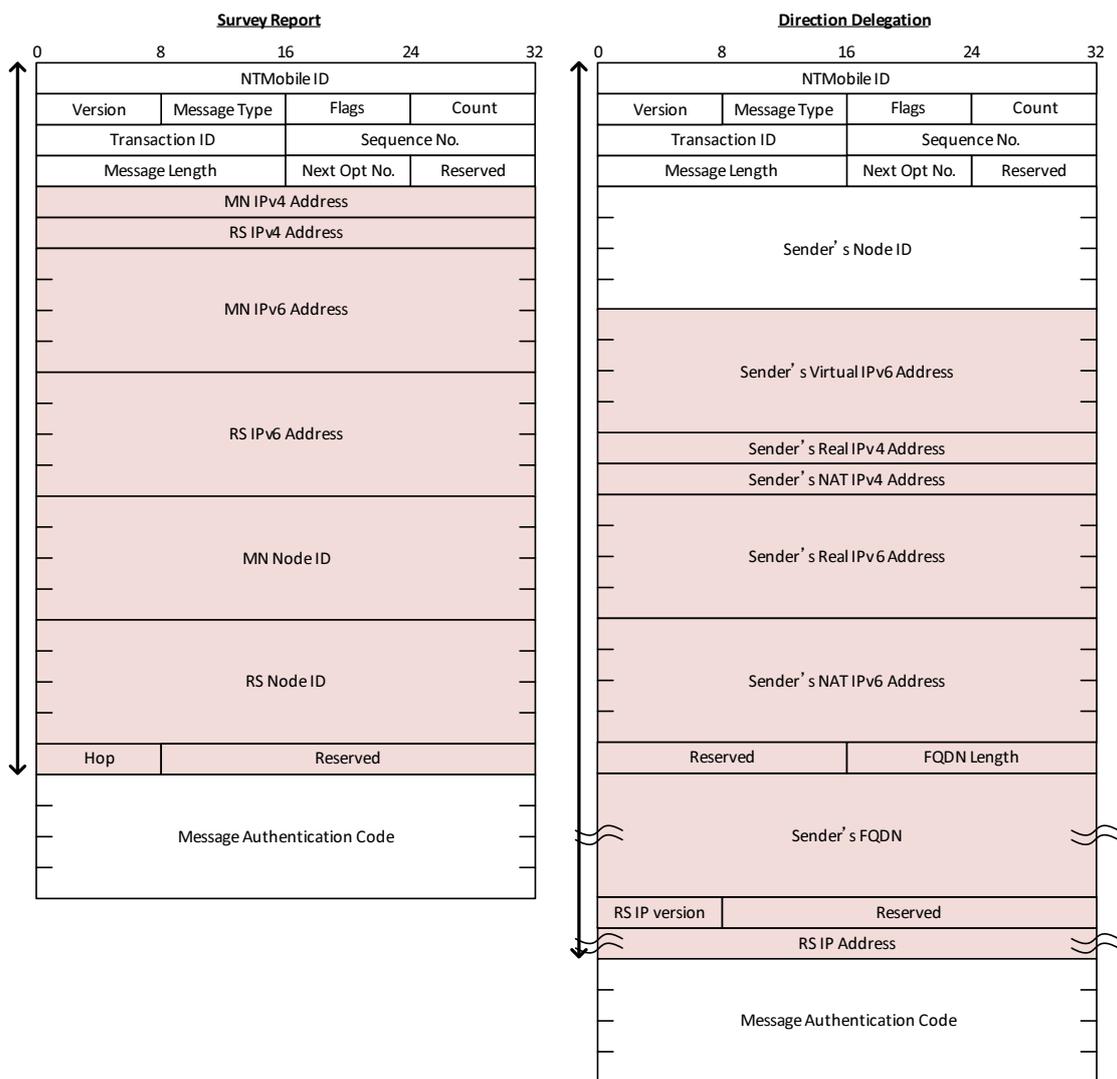


図 A.3 NTM 端末と RS のホップ数調査における提案のメッセージフォーマット 2

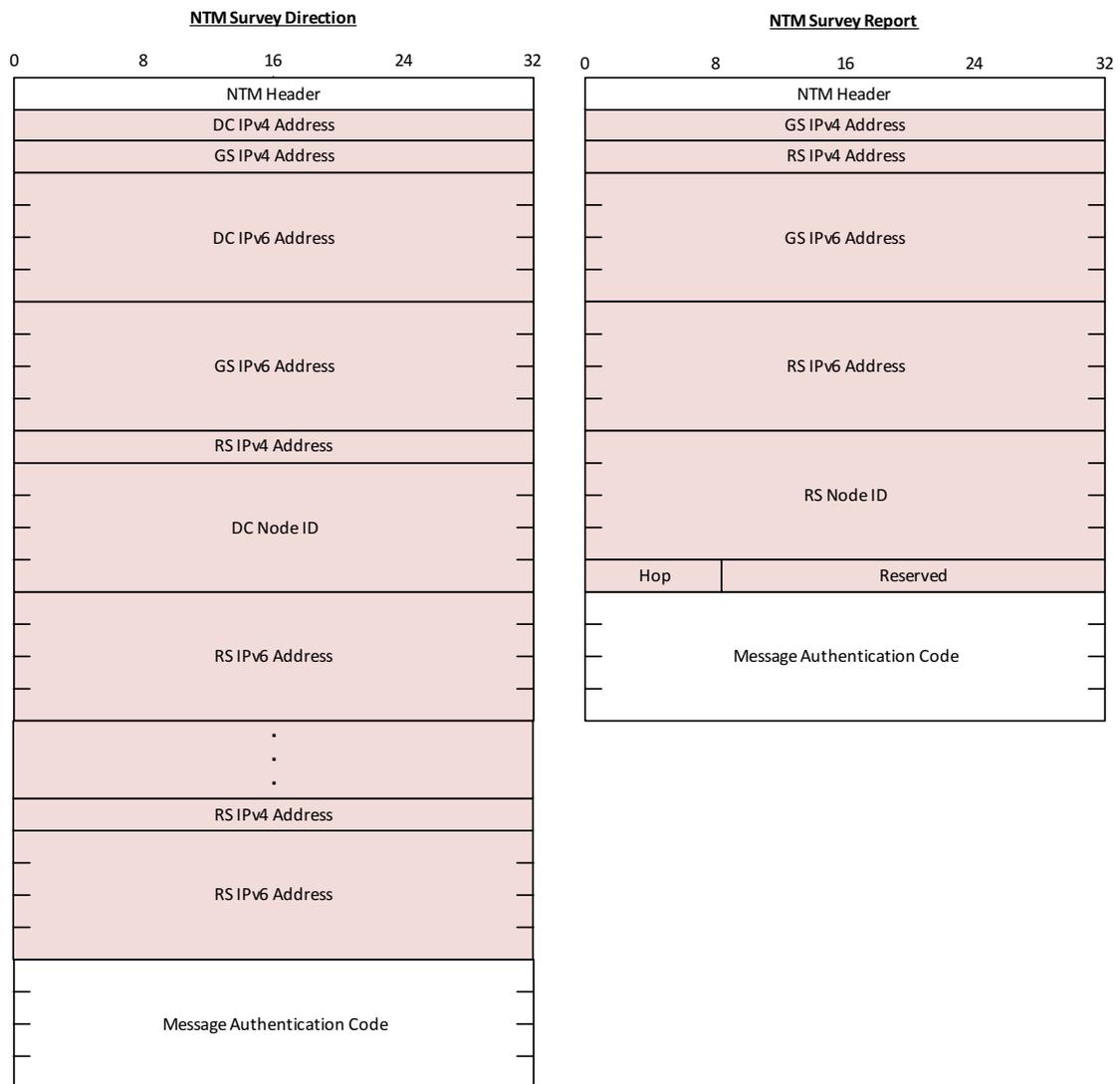


図 A.4 GS と RS 間のホップ数調査における提案のメッセージフォーマット

## 付録B 各OSのTTL初期値

表 B.1 に、主な OS の TTL の初期値を示す [19].

現在、主に利用されている OS の TTL の初期値は 64, 128, 255 である。また、実環境においてホップ数は 20 ホップ程度で収まる。その為、TTL 変化後の値から TTL の初期値を 64, 128, 255 のうちどれであるかを推測する。

表 B.1 各 OS の TTL 初期値

OS	Version	TTL
Linux	2.0.x kernel	64
Linux	2.2.14 kernel	255
Linux	2.4 kernel	255
Linux	Red hat 9	64
MacOS	2.0x	60
MacOS	X(10.5.6)	64
Windows	XP	128
Windows	7	128